

Contents

A Datasheet for Datasets	2
A.1 Motivation	2
A.2 Composition	2
A.3 Collection Process	3
A.4 Preprocessing/cleaning/labeling	3
A.5 Uses	3
A.6 Distribution	4
A.7 Maintenance	4
B Additional Information of ReinAD Dataset	5
B.1 Additional Details on Huan-in-the-loop Annotation Pipeline	5
B.2 Additional Visualization	5
C Additional Information of ReinADNet Method	28
C.1 Overview of Our Approach	28
C.2 Pyramidal Cost Aggregation Module	28
C.3 Patch-level Comparison Strategy	29
C.4 Predictor	30
C.5 Training and Inference	30
D Additional Information of Experiments	31
D.1 Detailed Quantitative Results for Each Category	31
D.2 Additional Qualitative Results	31

A Datasheet for Datasets

A.1 Motivation

- **For what purpose was the dataset created?**
Recent advances in industrial anomaly detection (IAD) benefit from existing datasets. However, the large performance gap between these benchmarks and real industrial practice reveals critical limitations. We argue that the mismatch between current datasets and real industrial scenarios becomes the primary barrier to practical IAD deployment. To bridge this gap, we propose ReinAD dataset, a comprehensive contrastive dataset towards real-world industrial anomaly detection.
- **Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?**
The authors of this paper.
- **Who funded the creation of the dataset?**
This work was supported by the National Natural Science Foundation of China. Detailed funding information will be updated upon acceptance.

A.2 Composition

- **What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?**
Our dataset consists of a training set and a test set. All normal and anomaly images are in JPG format, while all masks are in PNG format. In the mask annotations, pixels with a value of 0 represent normal regions, and pixels with a value of 1 represent anomaly regions.
- **How many instances are there in total (of each type, if appropriate)?**
Our dataset contains 71,955 instances and 59 categories in total.
- **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?**
Our dataset contains all possible instances.
- **What data does each instance consist of?**
For normal images, each instance consists of a single image in JPG format. For anomaly images, each instance consists of an image in JPG format and a pixel-level annotation file in PNG format.
- **Is there a label or target associated with each instance?**
The pixel-level label of anomalous areas is provided for each anomaly instance.
- **Is any information missing from individual instances?**
No.
- **Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?**
No.
- **Are there recommended data splits (e.g., training, development/validation, testing)?**
Our dataset is divided into a training set and a test set. The training set contains both normal and anomaly samples from seen categories, and the test set contains normal and anomaly samples from unseen categories. The category assignments are determined through random sampling to ensure unbiased evaluation.
- **Are there any errors, sources of noise, or redundancies in the dataset?**
The dataset are created and verified by human. The noise may come from human error in annotating data.
- **Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?**
The dataset is self-contained.
- **Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)?**
A subset of the dataset contains confidential data.

- **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?**
No.

A.3 Collection Process

- **How was the data associated with each instance acquired?**
Our data originated from multi-year accumulation in real industrial scenarios, with annotations created by certified professionals.
- **What mechanisms or procedures were used to collect the data (e.g., hardware apparatuses or sensors, manual human curation, software programs, software APIs)?**
We developed customized optical solutions tailored for different workpieces and anomaly types (e.g. low-angle ring light for scratches and multi-zone light for dents), ensuring comprehensive coverage across diverse scenarios and production lines. Technicians then defined anomaly criteria based on actual quality requirements and industrial SOP standards. During production, large quantities of both normal and anomaly samples were automatically captured, and subsequently labeled by annotators.
- **Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?**
The data was created by researchers and annotators. The annotators were paid about \$ 40 per day.
- **Over what timeframe was the data collected?**
The data was constructed from 2022 to 2025.
- **Were any ethical review processes conducted (e.g., by an institutional review board)?**
Yes, all data collection followed ethical guidelines. Anonymization was applied to sensitive samples.

A.4 Preprocessing/cleaning/labeling

- **Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?**
We excluded particularly blurred instances and those unrelated to the regions of interest. Pixel-level annotations were applied to all anomalous images.
- **Was the “raw” data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?**
Yes, actually the raw data in our dataset are all normal and anomaly images.
- **Is the software that was used to preprocess/clean/label the data available?**
VSCode was used to preprocess the data, and Labelme was used to label the data.

A.5 Uses

- **Has the dataset been used for any tasks already?**
No
- **Is there a repository that links to any or all papers or systems that use the dataset?**
No
- **What (other) tasks could the dataset be used for?**
Our dataset could be used for training and evaluating industrial anomaly detection models, such as supervised, semi-supervised and unsupervised anomaly detection tasks
- **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?**
No.
- **Are there tasks for which the dataset should not be used?**
No.
- **Are there any potential negative social impacts?**
No.

A.6 Distribution

- **Will the dataset be distributed to third parties outside of the entity (*e.g.*, company, institution, organization) on behalf of which the dataset was created?**
No.
- **How will the dataset will be distributed (*e.g.*, tarball on website, API, GitHub)?**
The dataset will be released at Kaggle and Github.
- **Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?**
The dataset will be released under the Apache License.
- **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?**
No.
- **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?**
No.

A.7 Maintenance

- **Who will be supporting/hosting/maintaining the dataset?**
The authors of this paper.
- **How can the owner/curator/manager of the dataset be contacted (*e.g.*, email address)?**
Please contact with authors through emails in the paper.
- **Is there an erratum?**
No.
- **Will the dataset be updated (*e.g.*, to correct labeling errors, add new instances, delete instances)?**
Yes, users can propose issues and the dataset will be updated on Github.
- **Will older versions of the dataset continue to be supported/hosted/maintained?**
Primarily, we plan to maintain only the most recent version of the dataset. However, under certain circumstances, such as significant updates to our dataset or the need for validation of previous research work using older versions, we will exceptionally preserve previous versions of the dataset for up to one year.
- **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?**
Contact the author of the paper.

Table 1: **Quantitative details** on annotation quality improvement with the human-in-the-loop annotation pipeline.

	Data (Pure human label)	Data (Mode label → human correct)	Human corrected area / GT mask area	Performance (IoU0.5)
Iteration 1	5%	0%	-	72.4%
Iteration 2	5%	10%	35%	85.6%
Iteration 3	5%	50%	12%	90.2%
Iteration 4	5%	95%	4.7%	93.8%

B Additional Information of ReinAD Dataset

B.1 Additional Details on Huan-in-the-loop Annotation Pipeline

Our human-in-the-loop annotation pipeline significantly improves annotation accuracy and reduces human effort through iterative refinement. Tab. 1 are quantitative details.

Initially, human experts annotate 5% of the data. The model trained on this dataset achieves 72.4% pixel-wise accuracy (IoU@0.5). During the second iteration, the model generates pre-annotations for 10% of the data, and then human experts correct 35% of the mask area pixels. The model trained on this expanded 15% dataset then improved to 85.6% accuracy. With each iteration, the amount of manual correction gradually decreases, while model performance consistently improves. In the final iteration, human experts only need to correct 4.7% of the ground truth mask pixels, and the model achieves quite high performance (93.8% IoU@0.5).

B.2 Additional Visualization

The additional visualization of all categories in our ReinAD dataset is shown from Fig. 1 to Fig. 45.

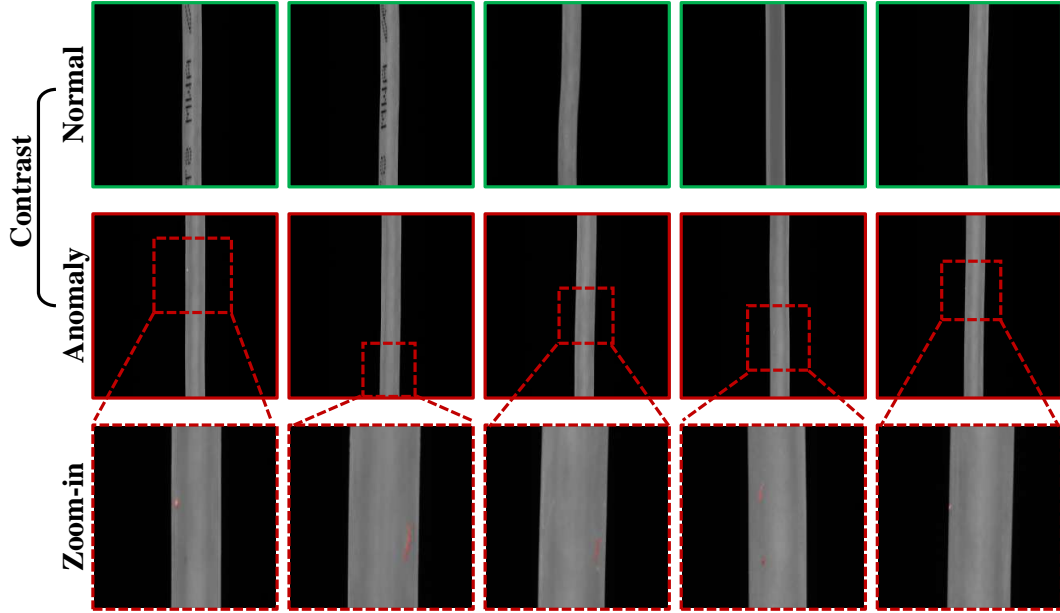


Figure 1: **Visualization of Cable 1 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

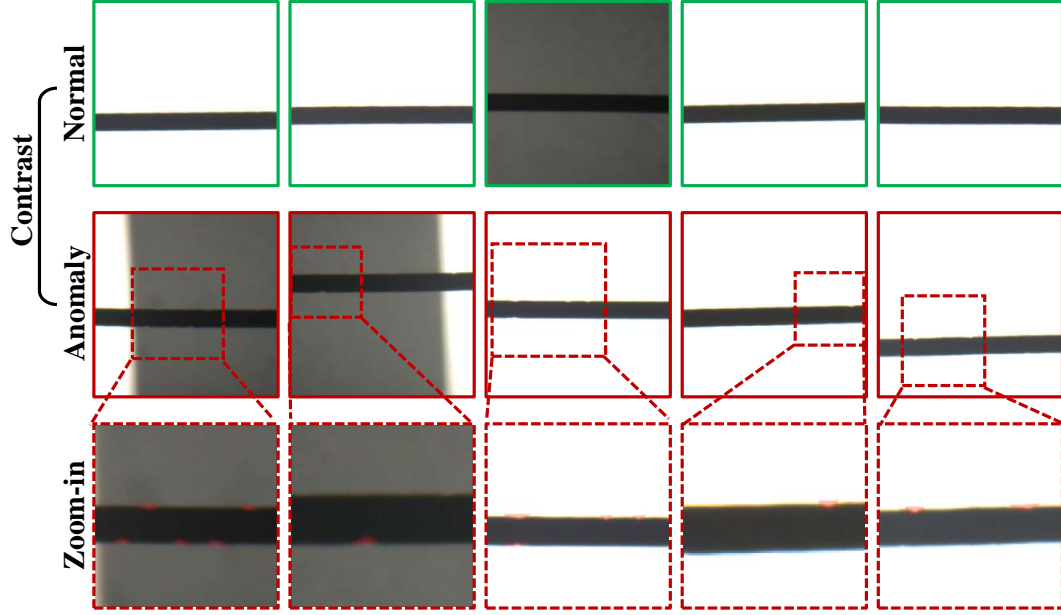


Figure 2: **Visualization of *Cable 2* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

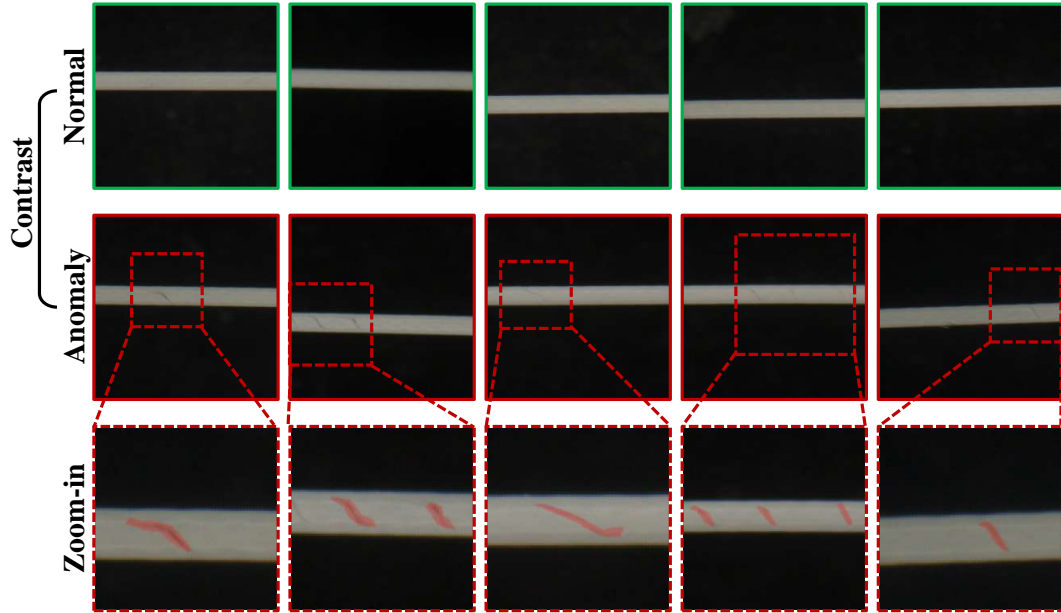


Figure 3: **Visualization of *Cable 3* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

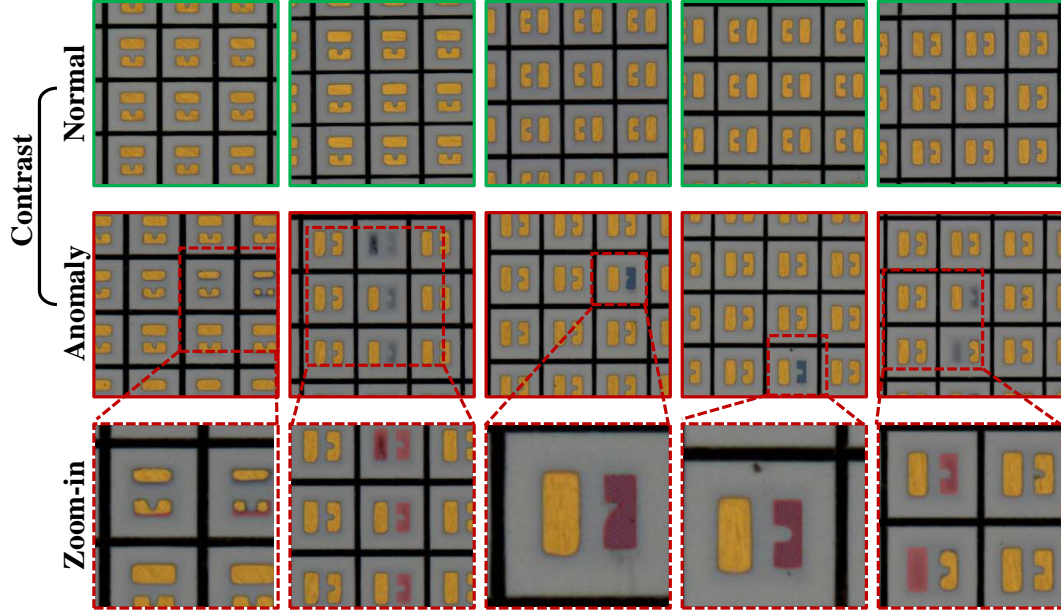


Figure 4: **Visualization of LED 1 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

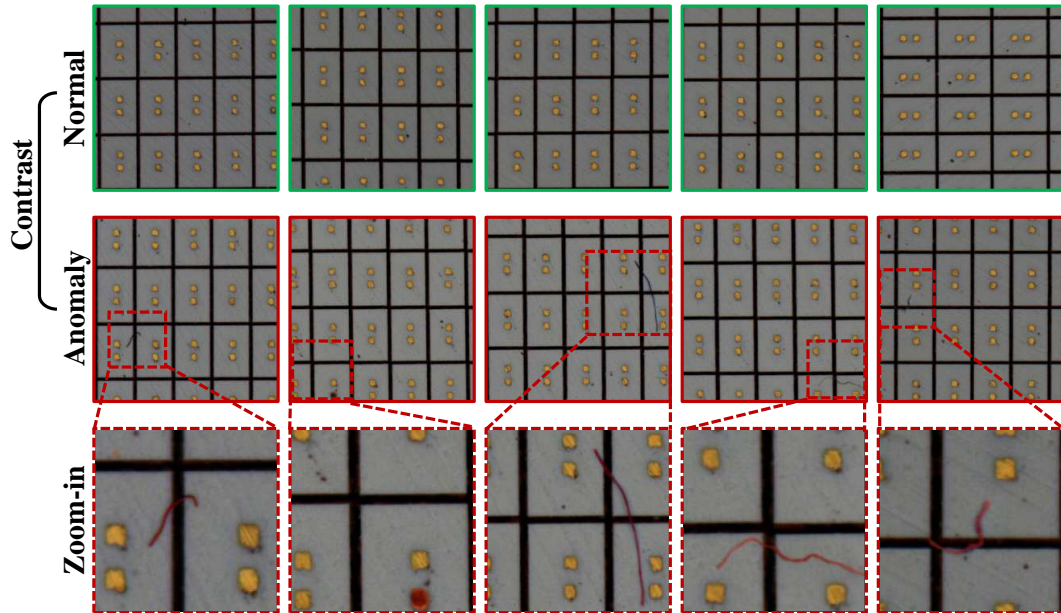


Figure 5: **Visualization of LED 2 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

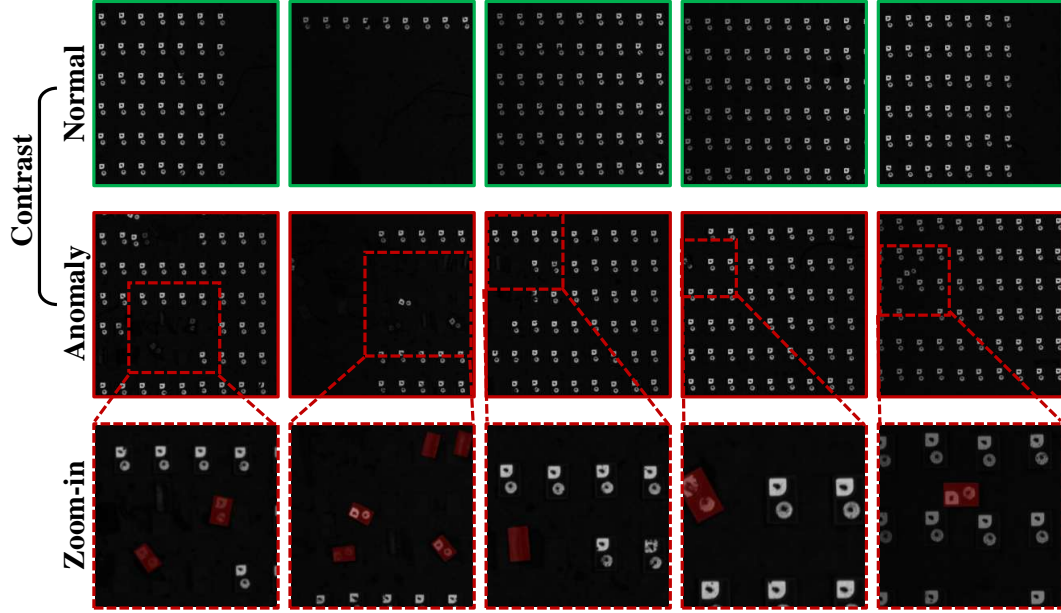


Figure 6: **Visualization of LED 3 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

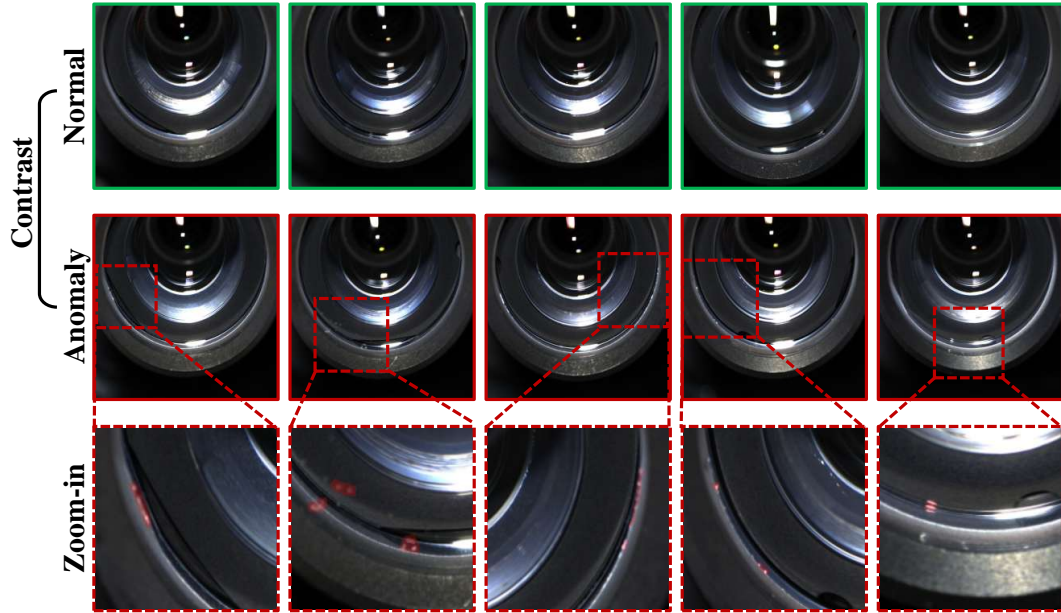


Figure 7: **Visualization of Lens 1 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

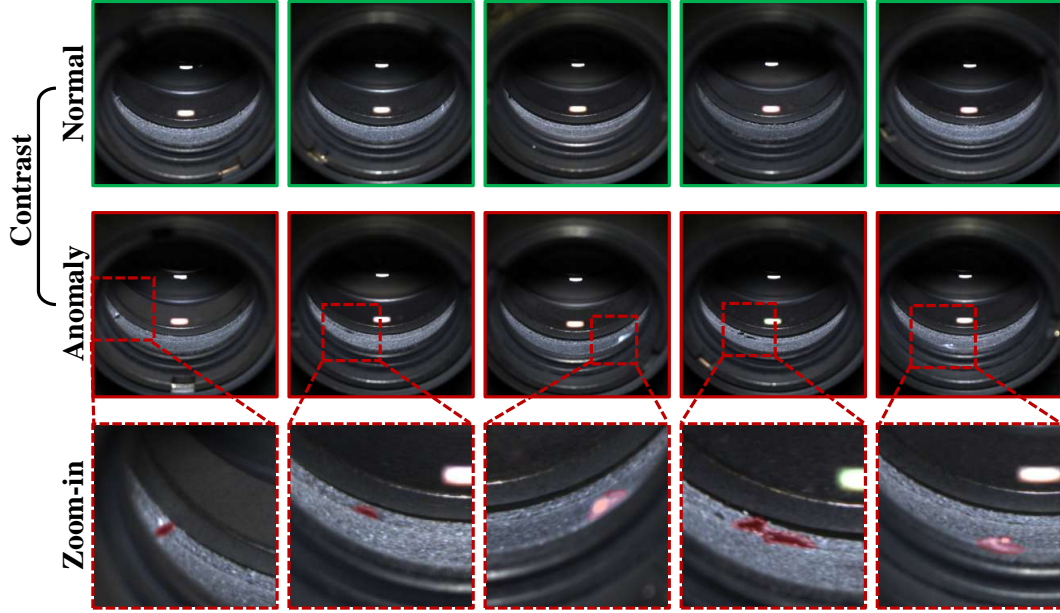


Figure 8: **Visualization of *Lens 2* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

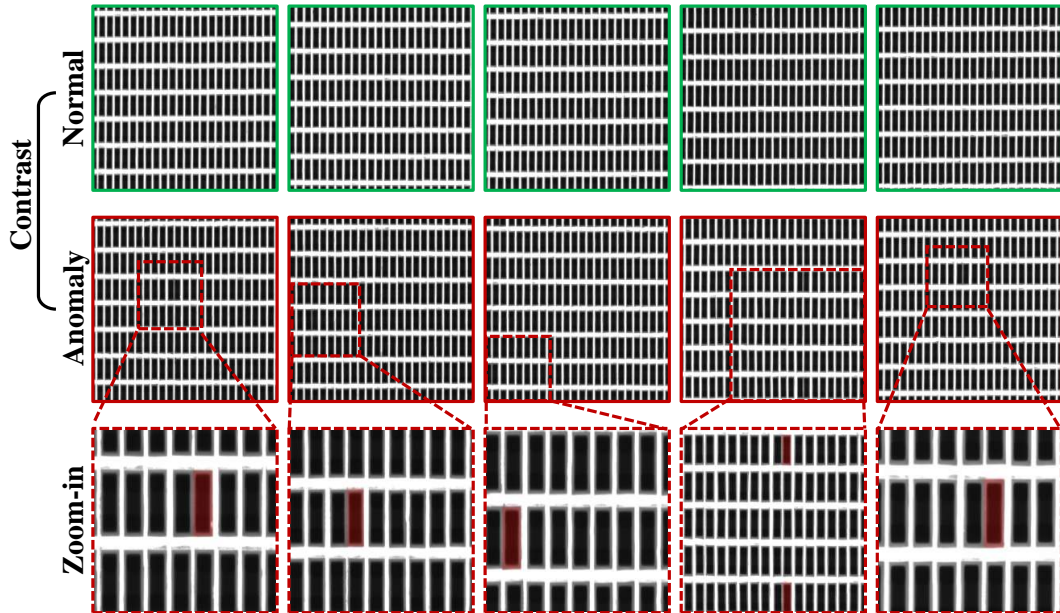


Figure 9: **Visualization of *MiniLED 1* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

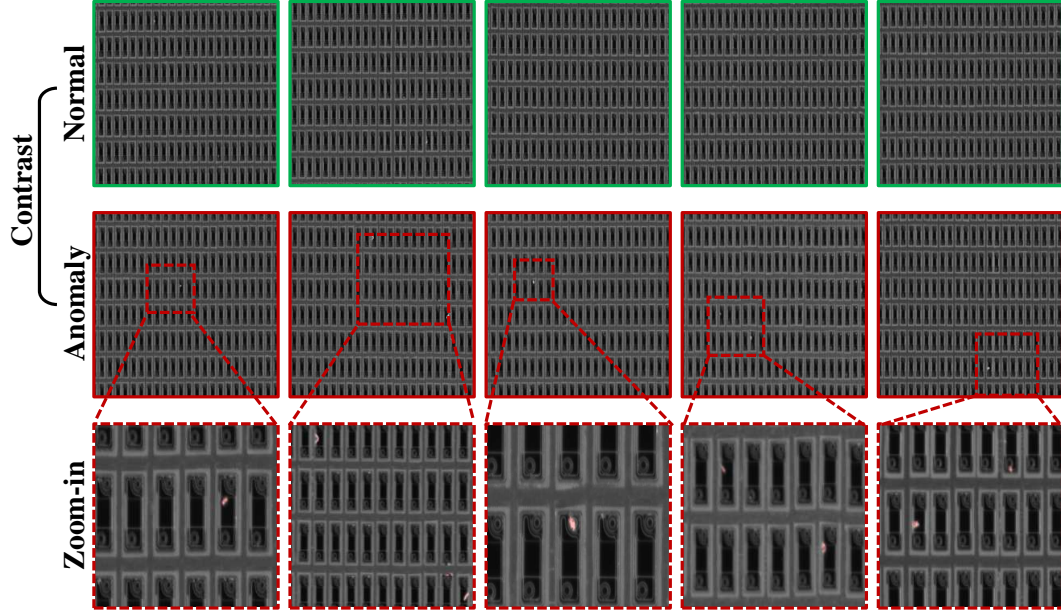


Figure 10: **Visualization of *MiniLED 2* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

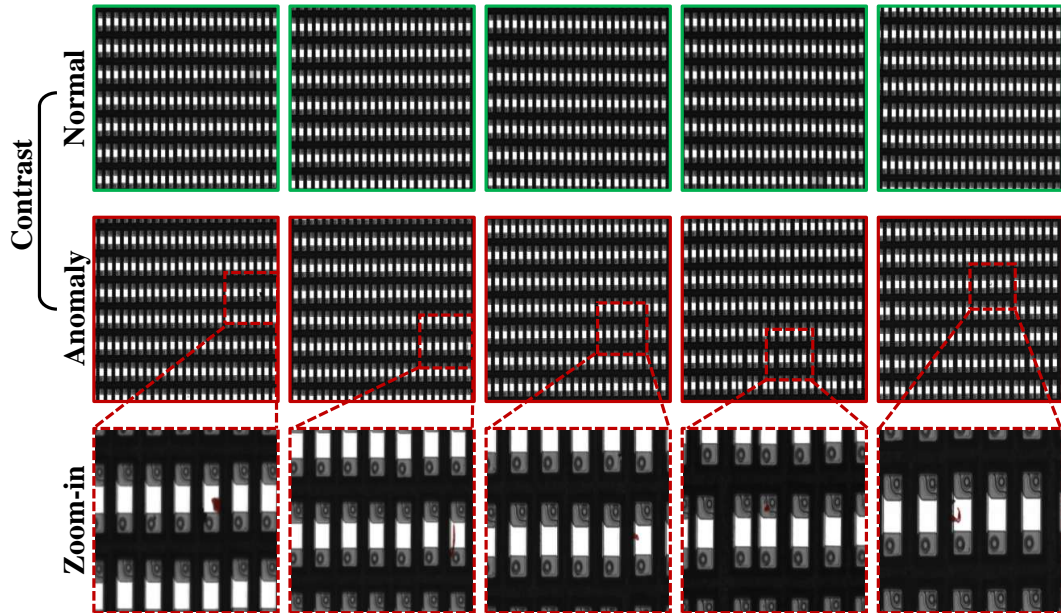


Figure 11: **Visualization of *MiniLED 3* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

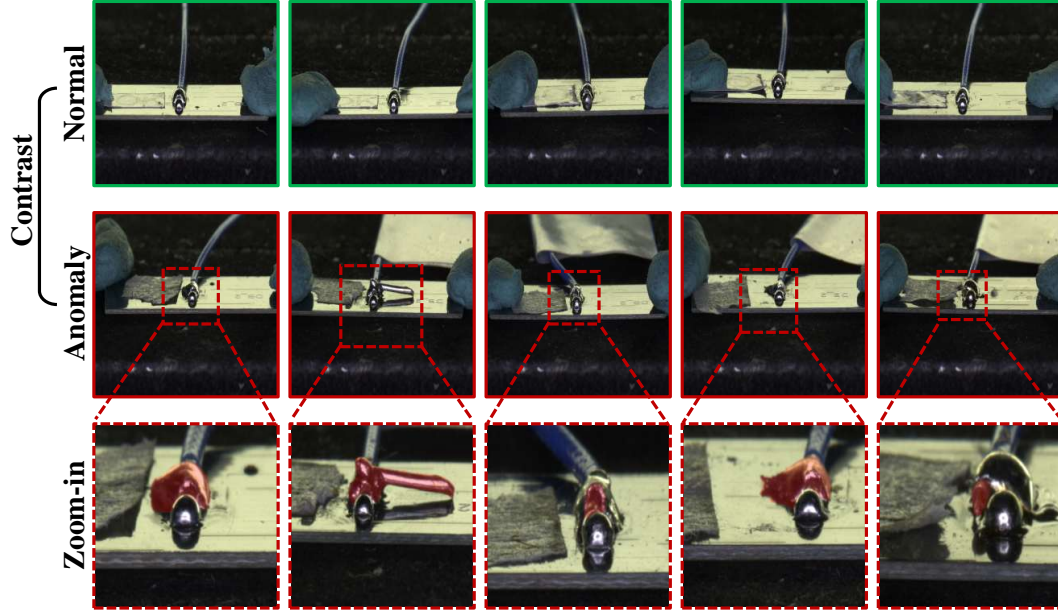


Figure 12: **Visualization of PCB Solder 1 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

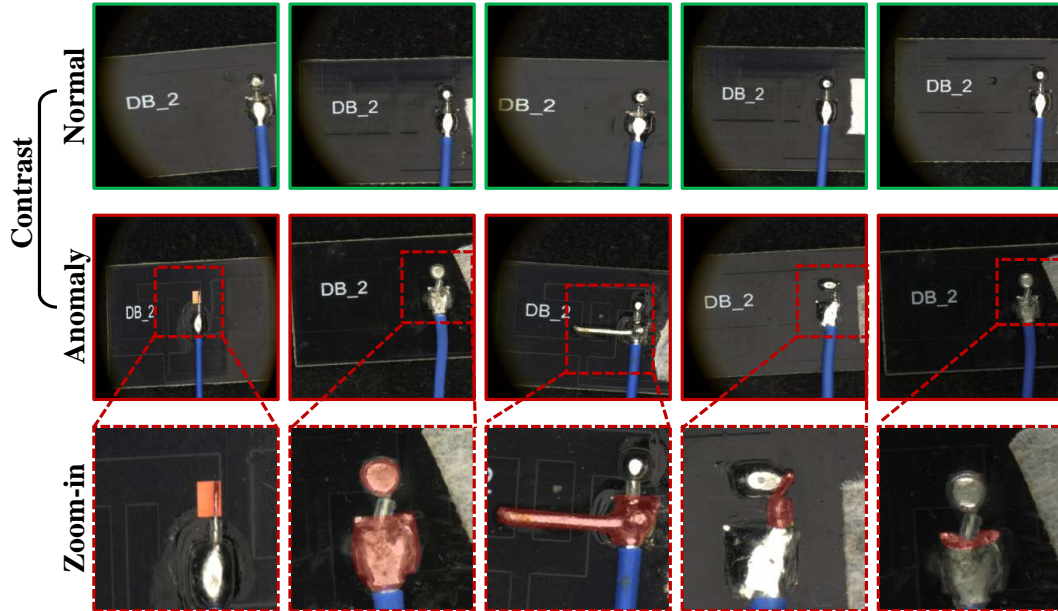


Figure 13: **Visualization of PCB Solder 2 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

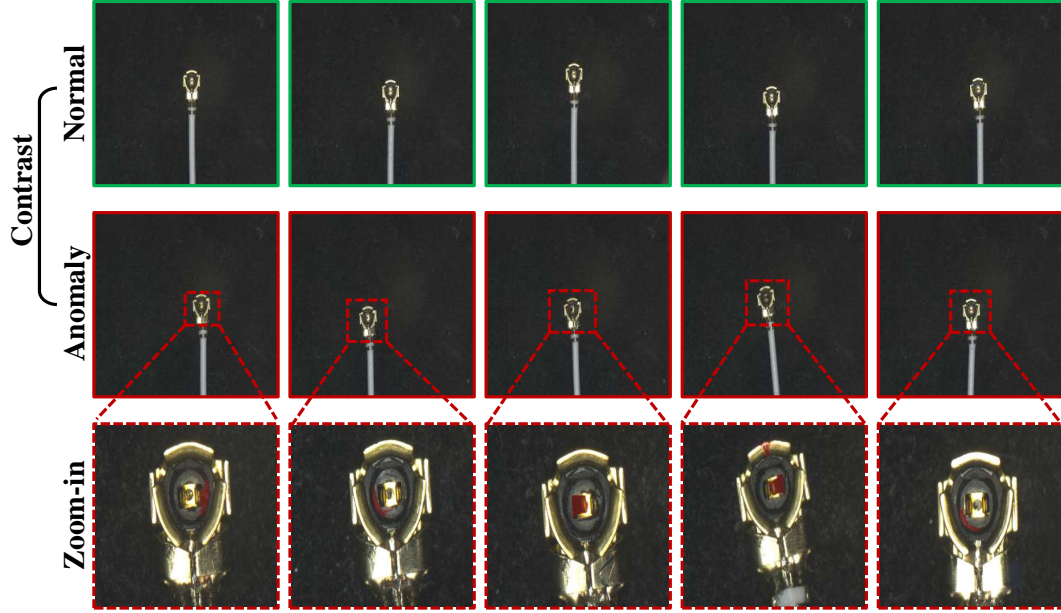


Figure 14: **Visualization of PCB Terminal in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

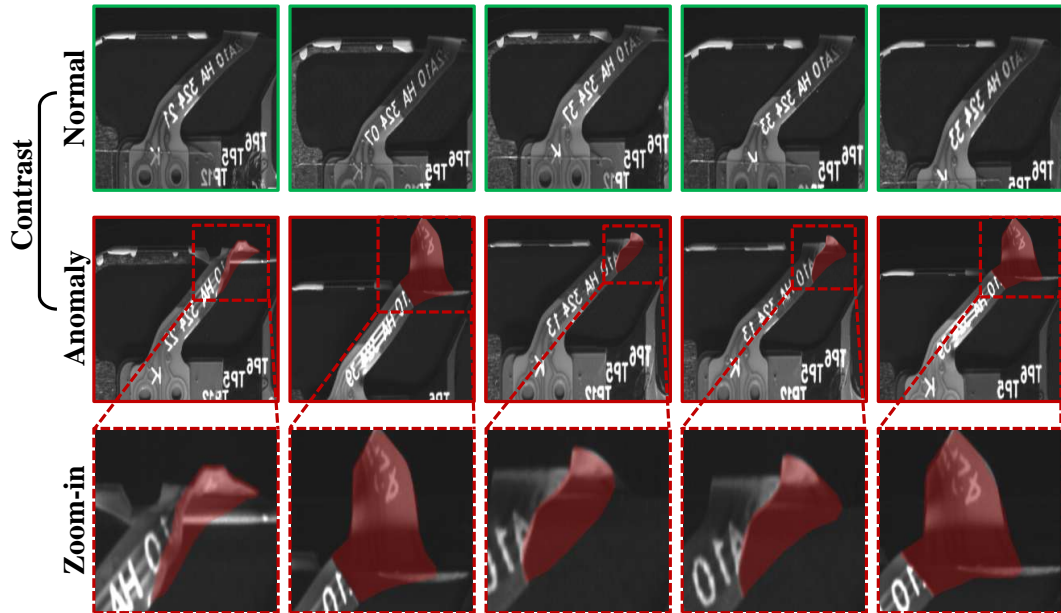


Figure 15: **Visualization of Pinline in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

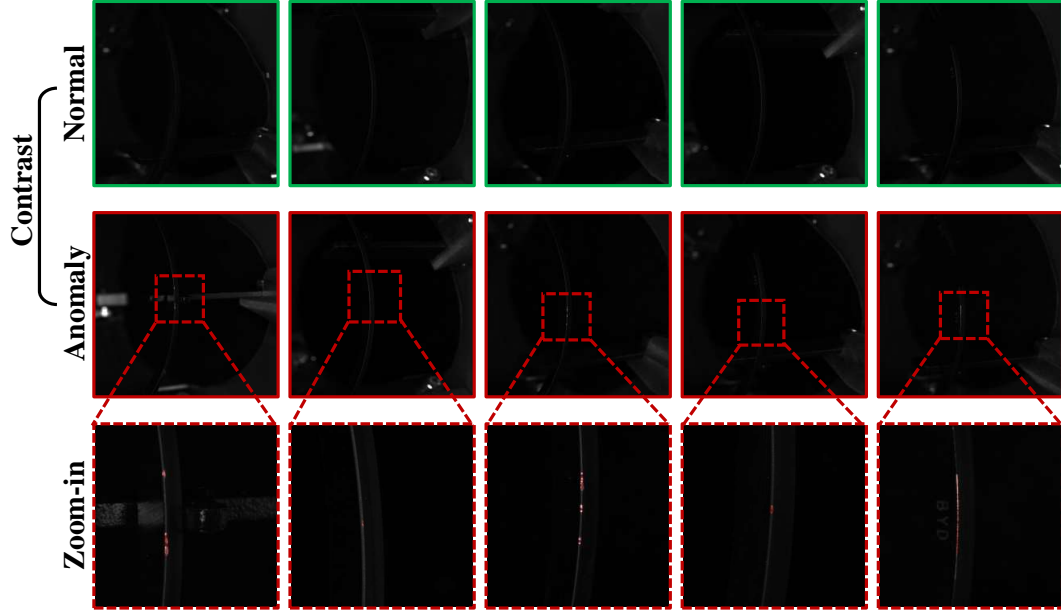


Figure 16: **Visualization of *Piston Ring 1* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

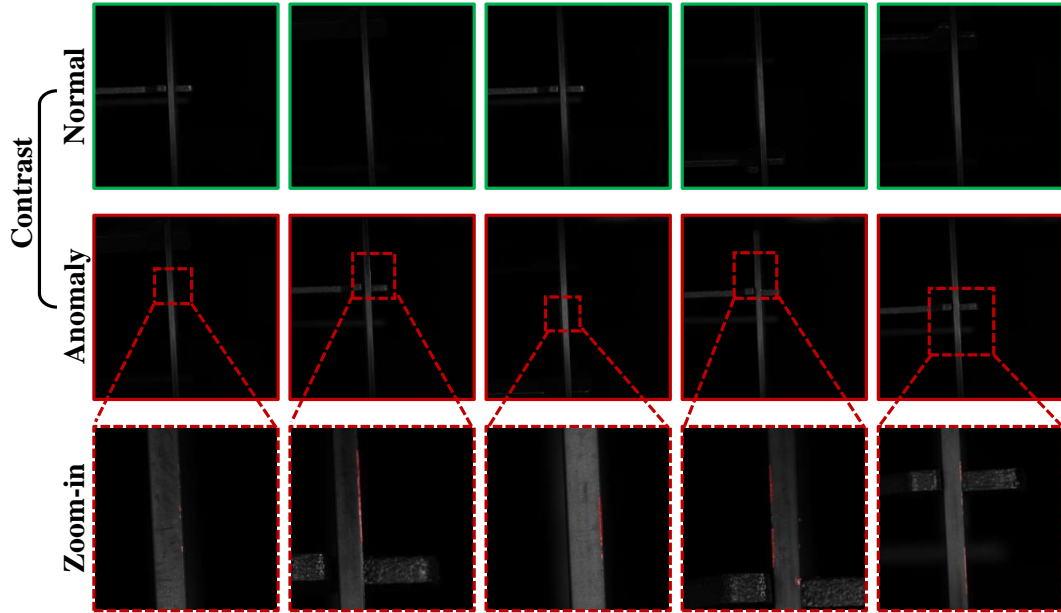


Figure 17: **Visualization of *Piston Ring 2* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

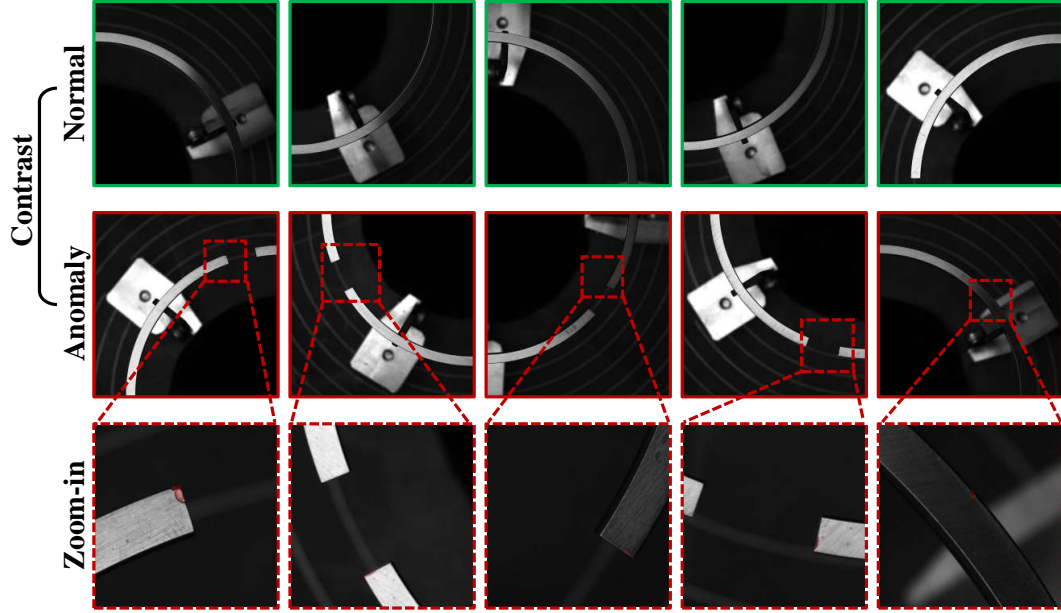


Figure 18: **Visualization of *Piston Ring 3* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

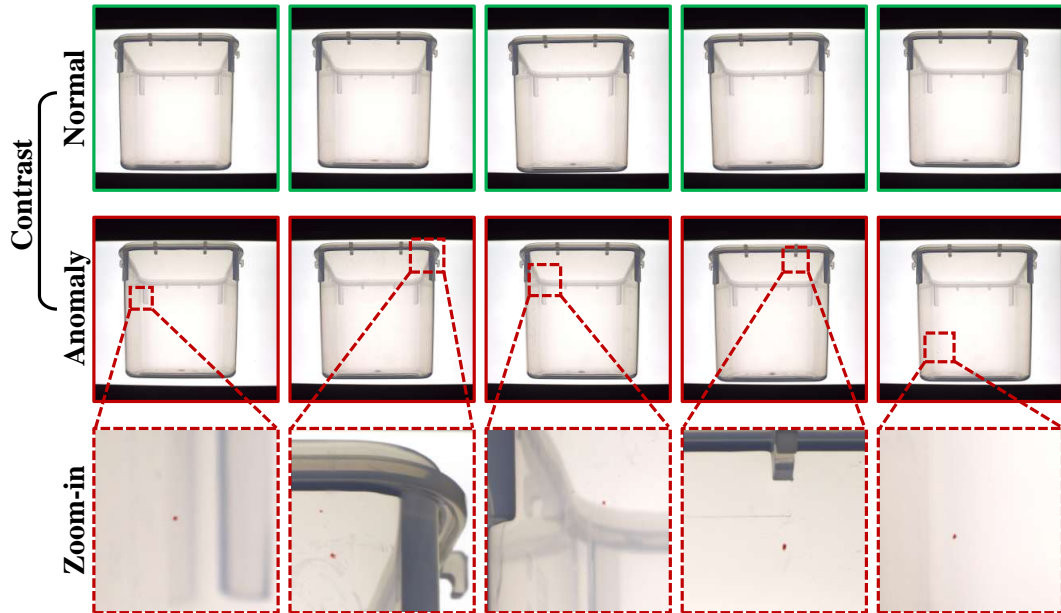


Figure 19: **Visualization of *Plastic Box* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

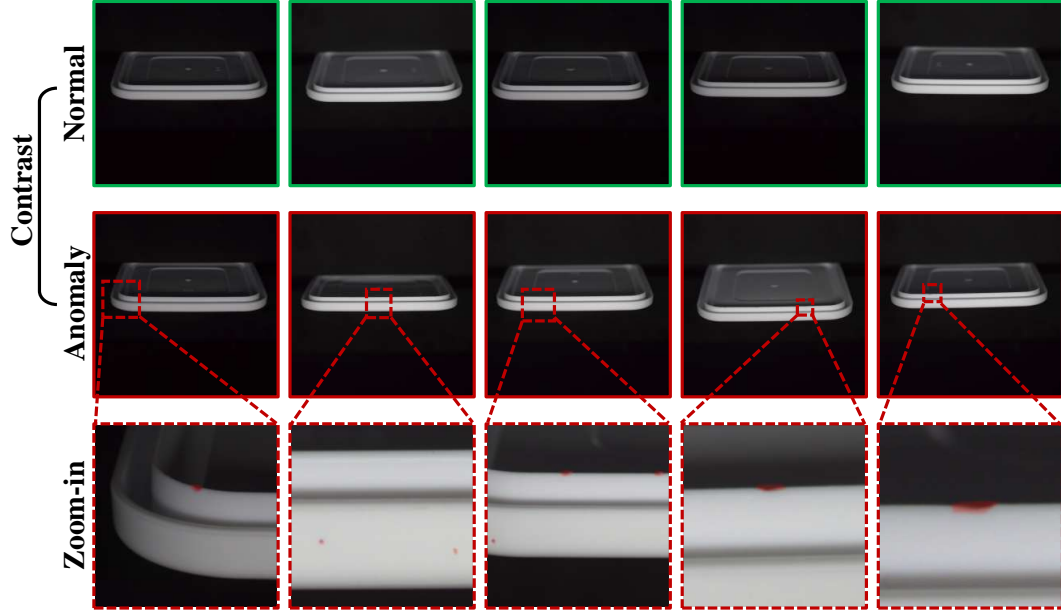


Figure 20: **Visualization of *Plastic Cover 1* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

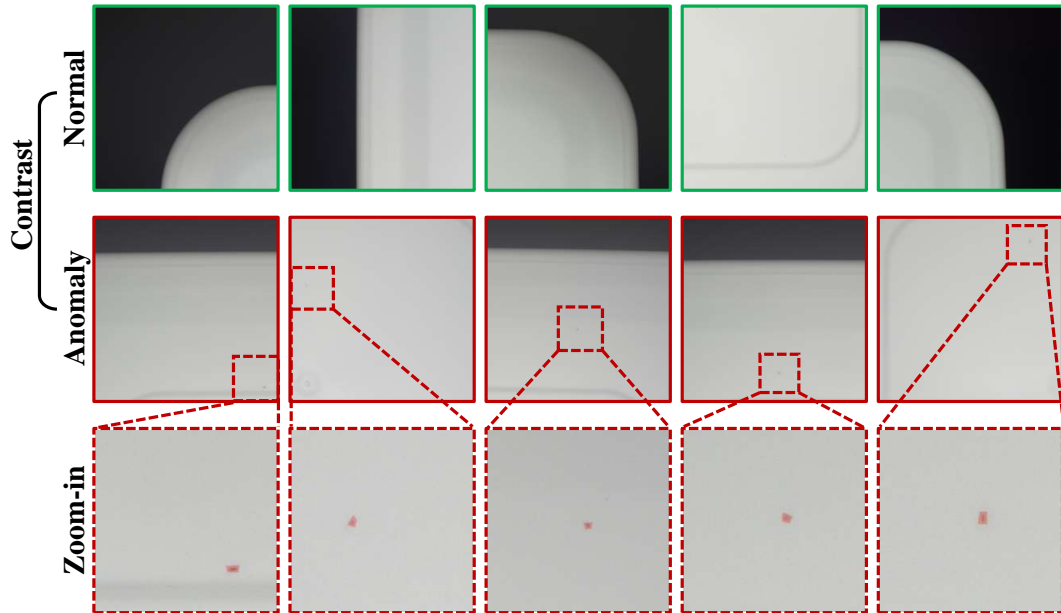


Figure 21: **Visualization of *Plastic Cover 2* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

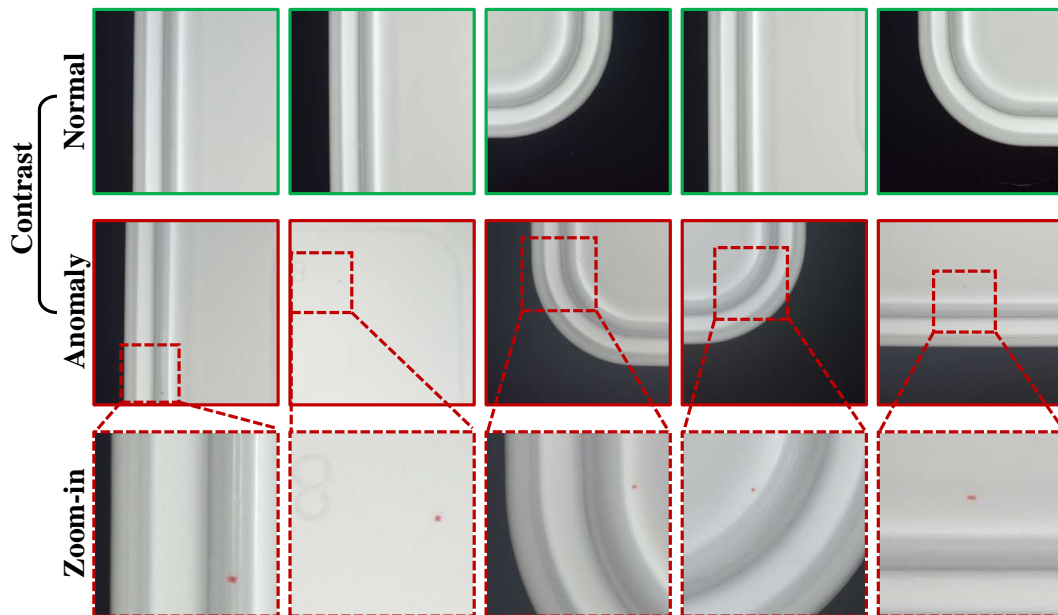


Figure 22: **Visualization of *Plastic Cover 3* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

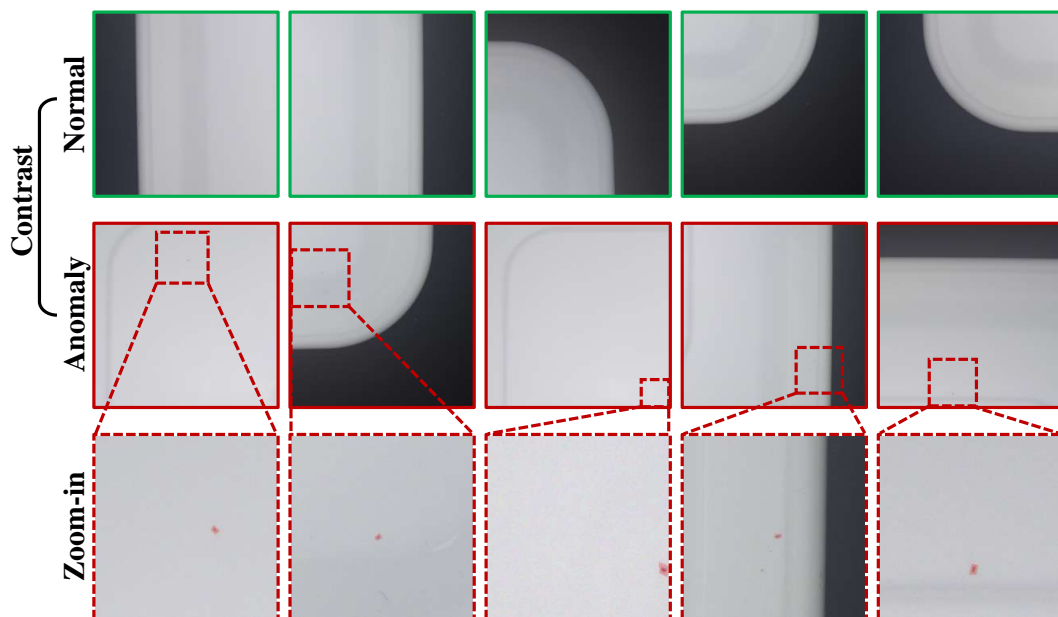


Figure 23: **Visualization of *Plastic Cover 4* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

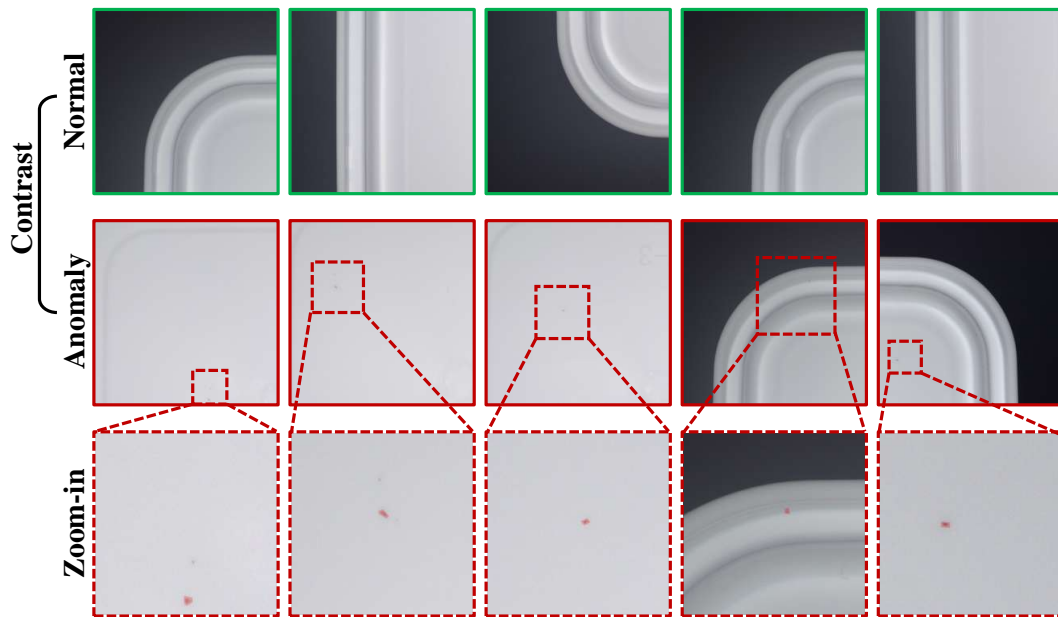


Figure 24: **Visualization of *Plastic Cover 5* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

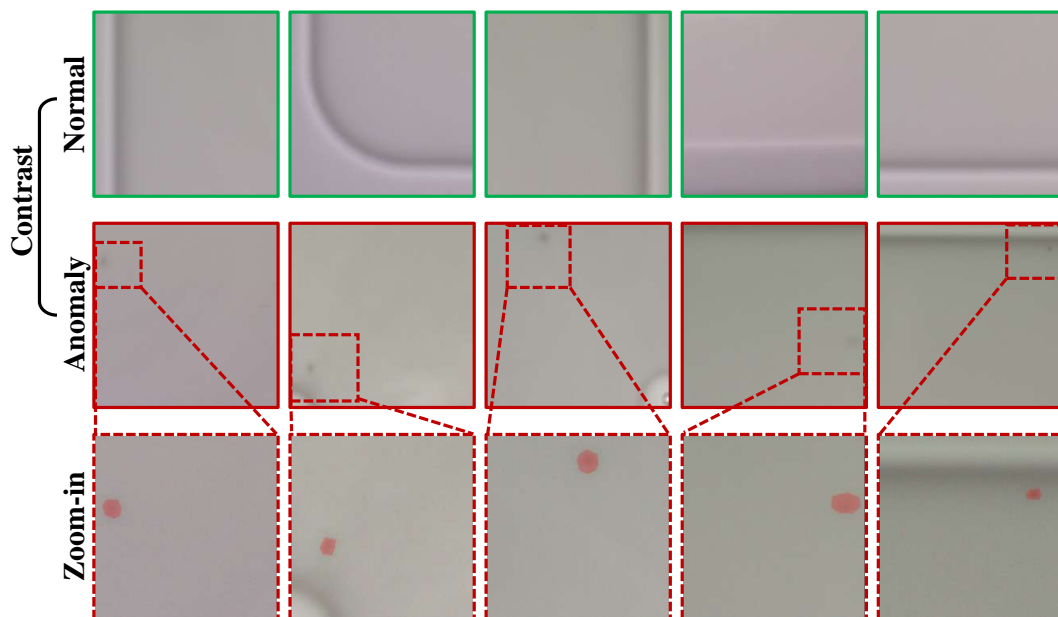


Figure 25: **Visualization of *Plastic Cover 6* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

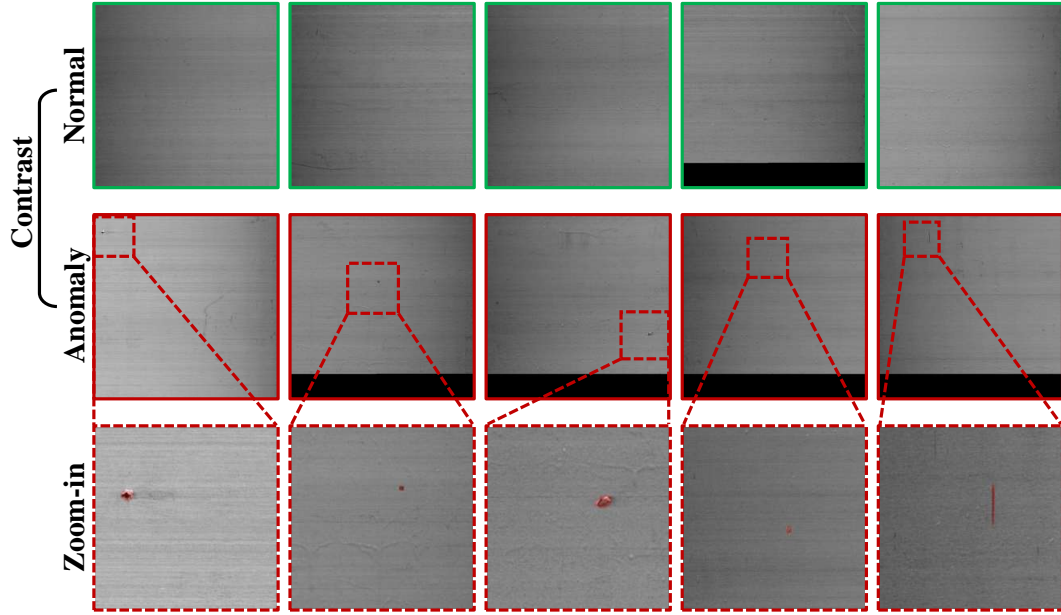


Figure 26: **Visualization of Profile Surface 1 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

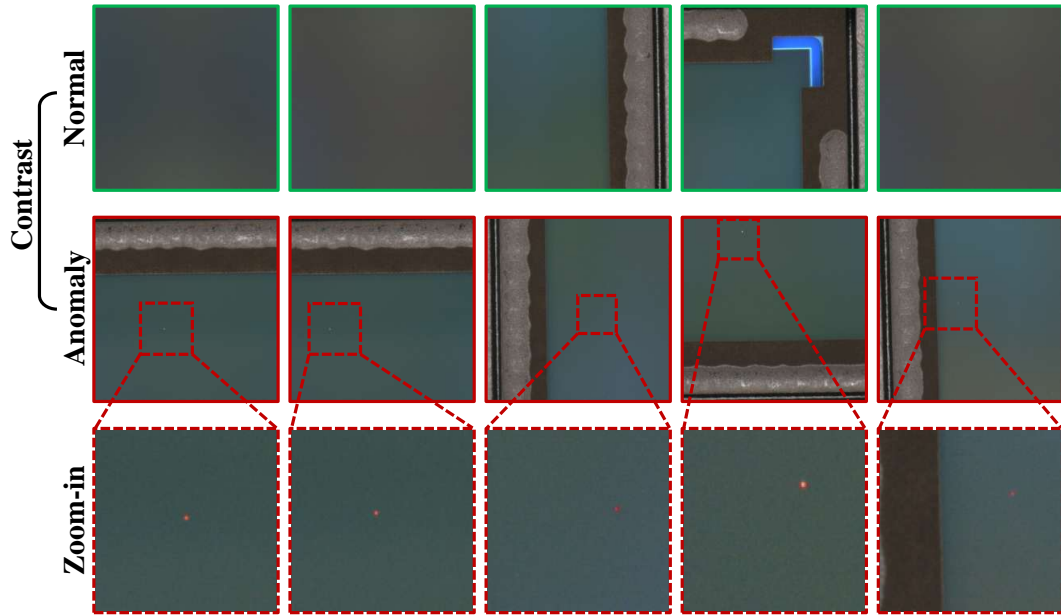


Figure 27: **Visualization of Profile Surface 2 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

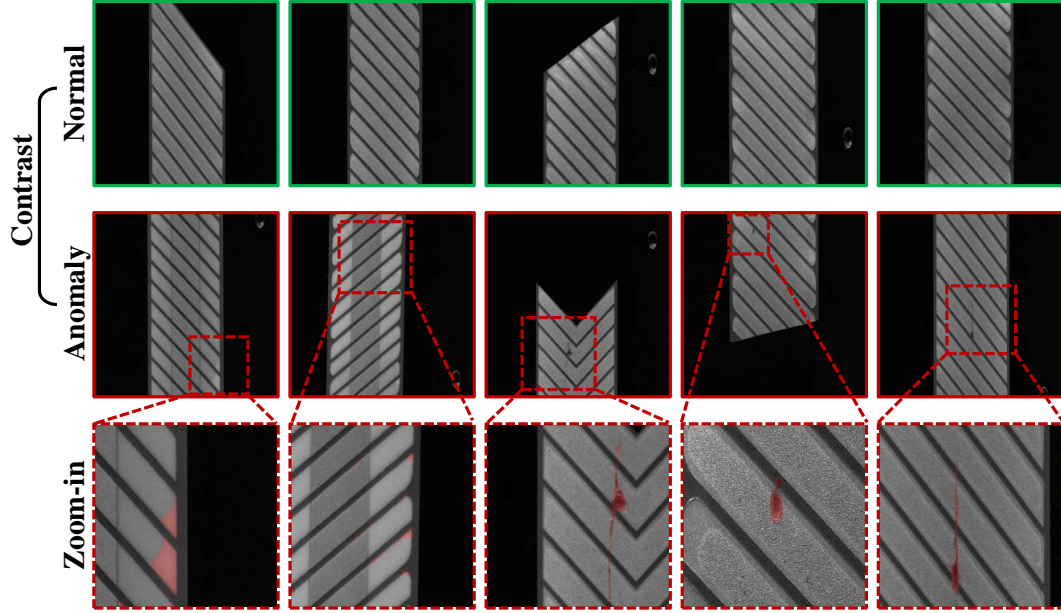


Figure 28: **Visualization of *Reflective Sheet* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

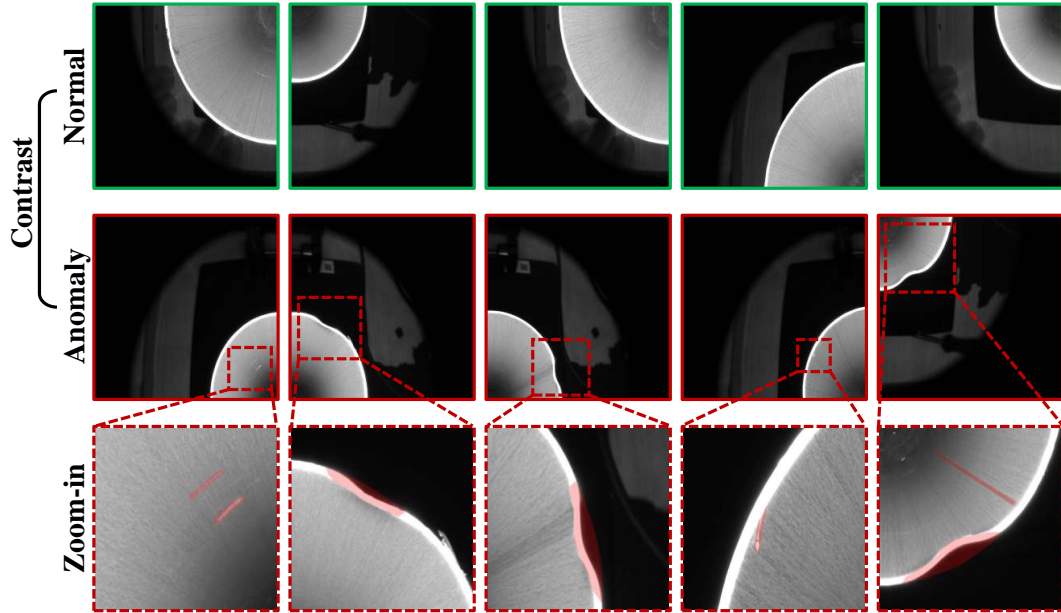


Figure 29: **Visualization of *Round Tube* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

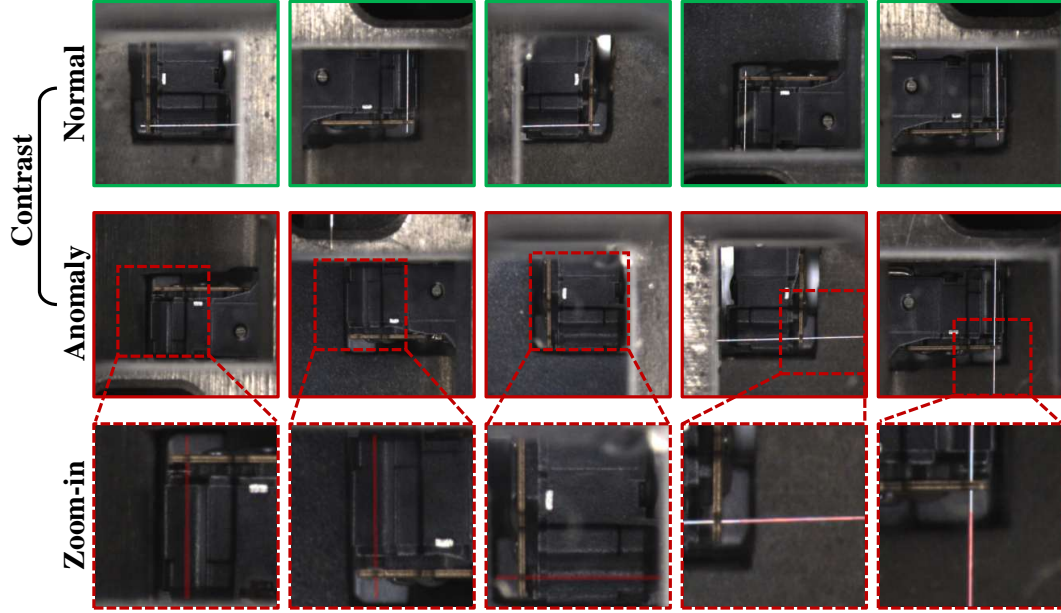


Figure 30: **Visualization of *Suspension Wire* in ReinAD dataset.** Top row: Normal images with **green** borders; Middle row: Anomaly images with **red** borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in **red** masks.

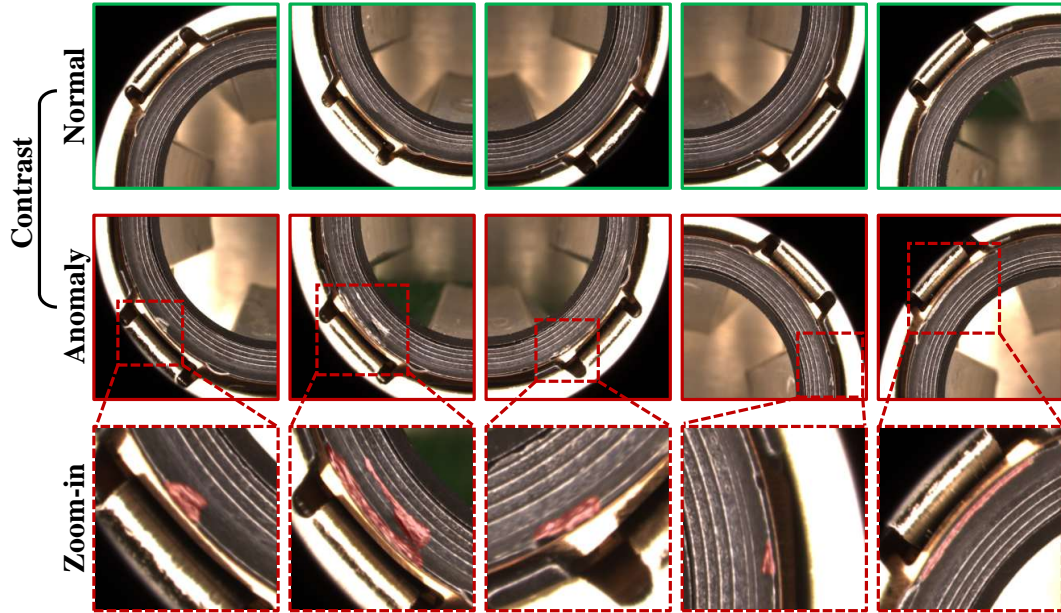


Figure 31: **Visualization of *Thread* in ReinAD dataset.** Top row: Normal images with **green** borders; Middle row: Anomaly images with **red** borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in **red** masks.

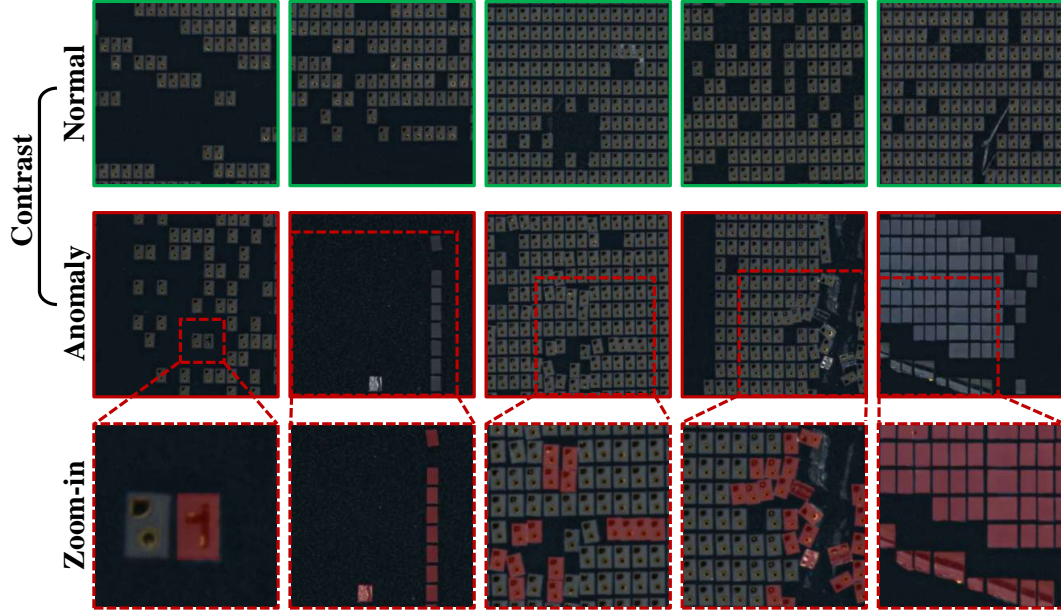


Figure 32: **Visualization of Wafer 1 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

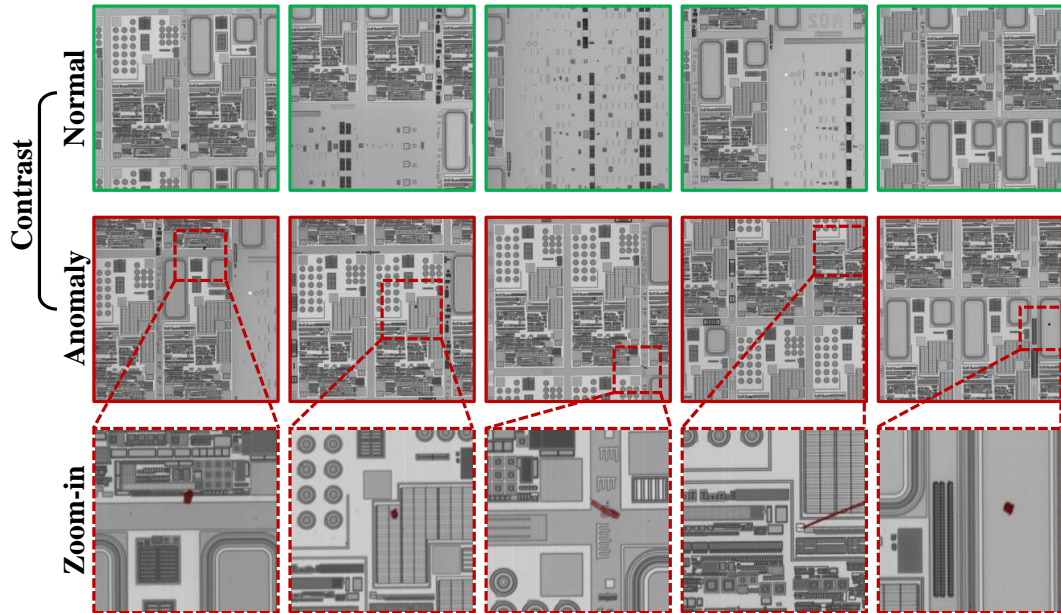


Figure 33: **Visualization of Wafer 2 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

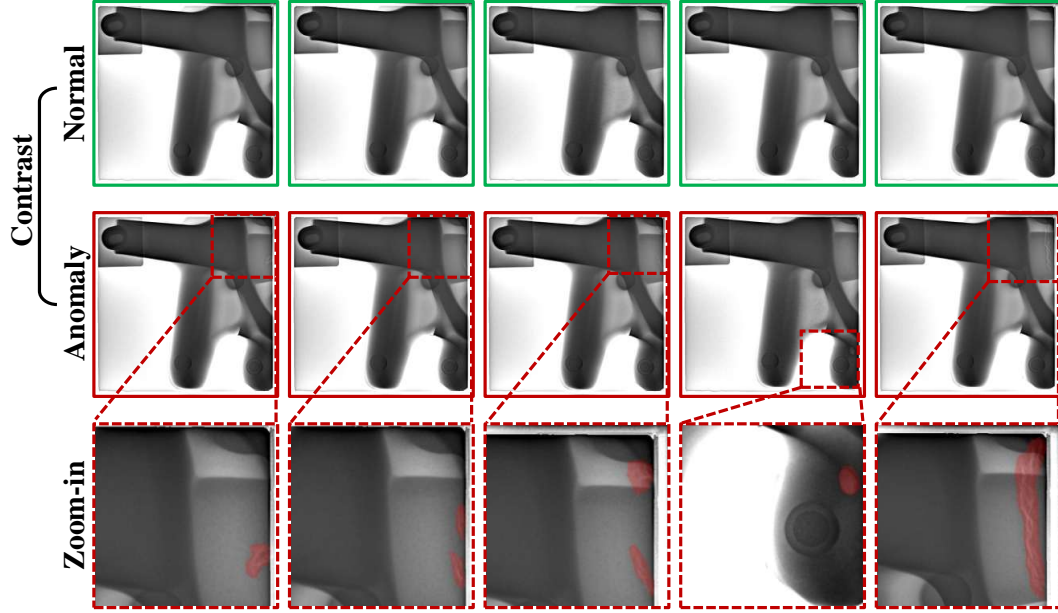


Figure 34: **Visualization of *Bearing 1* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

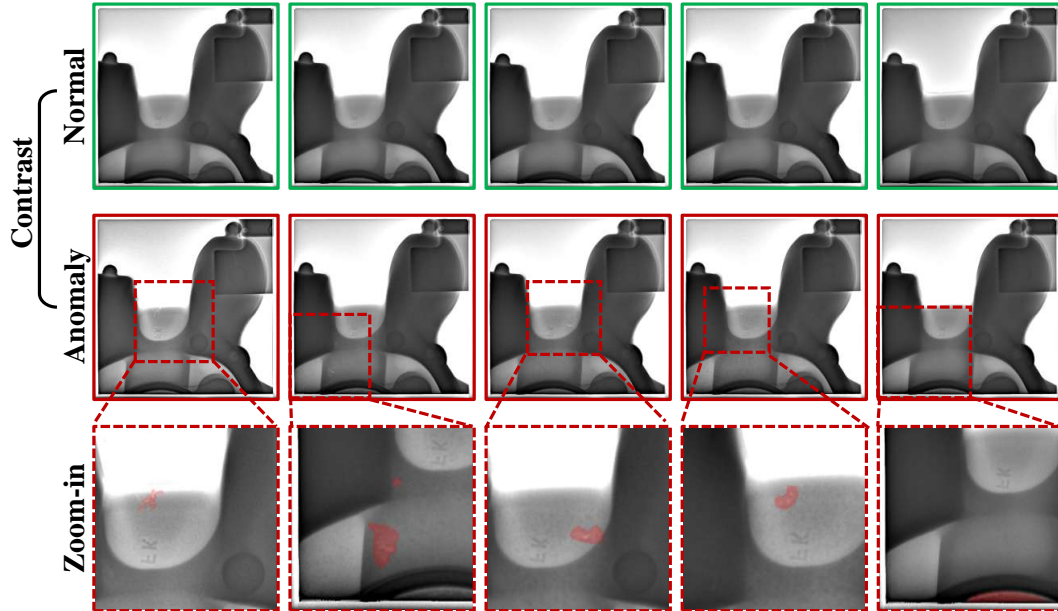


Figure 35: **Visualization of *Bearing 2* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

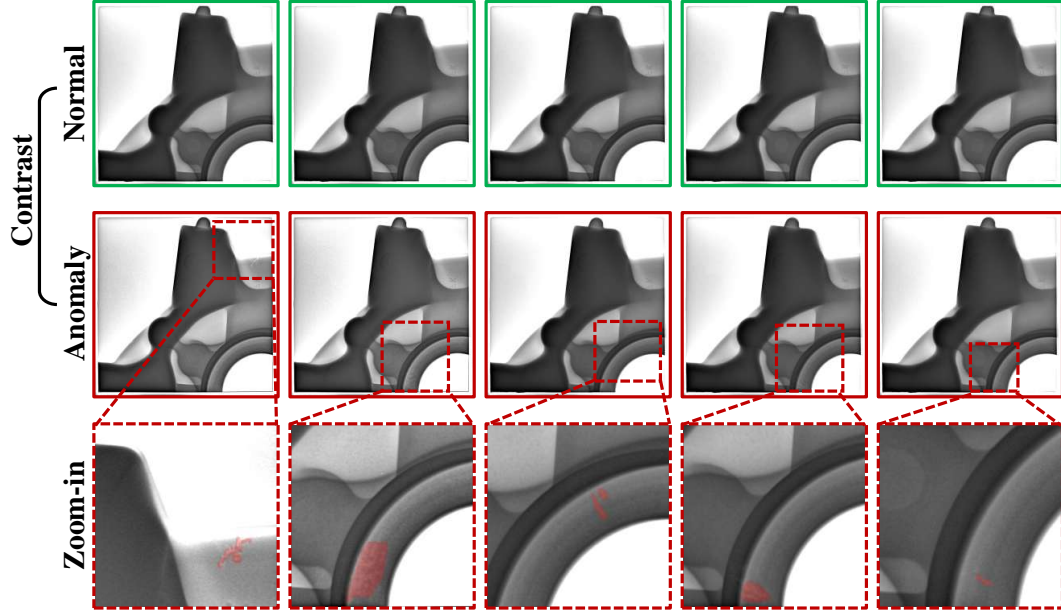


Figure 36: **Visualization of *Bearing 3* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

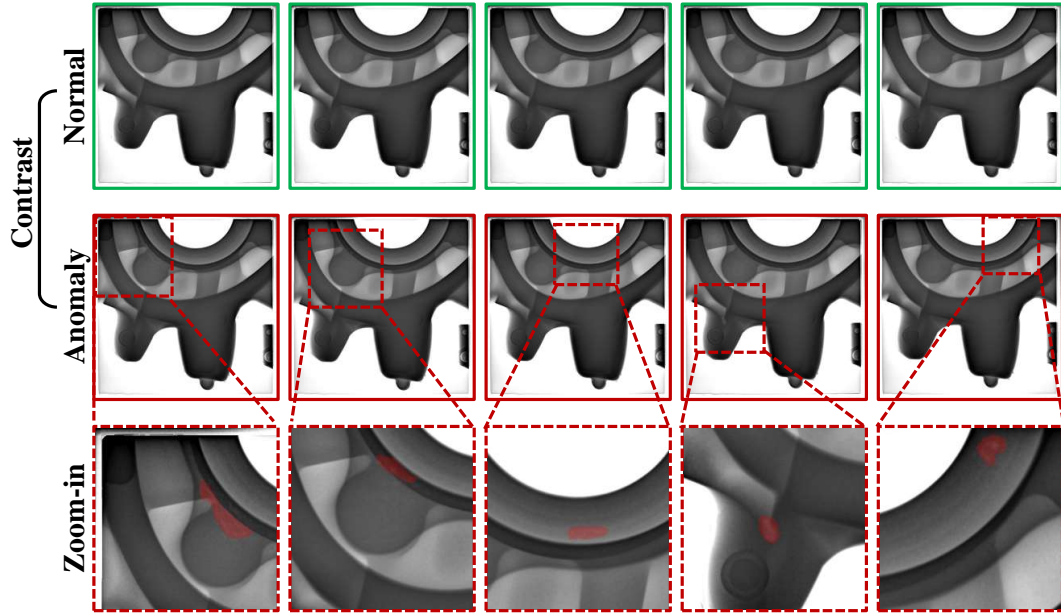


Figure 37: **Visualization of *Bearing 4* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

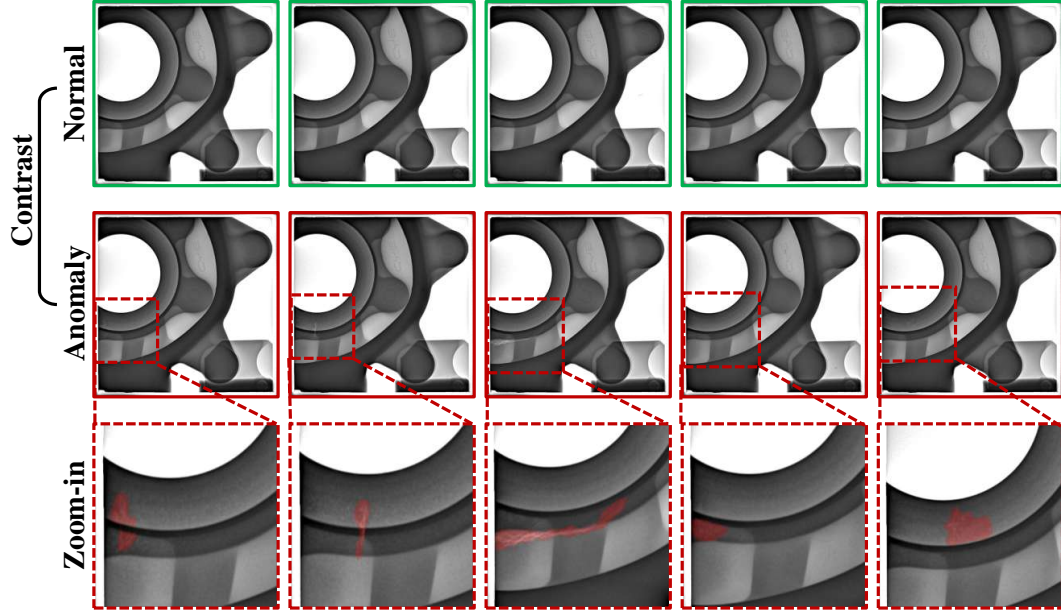


Figure 38: **Visualization of *Bearing 5* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

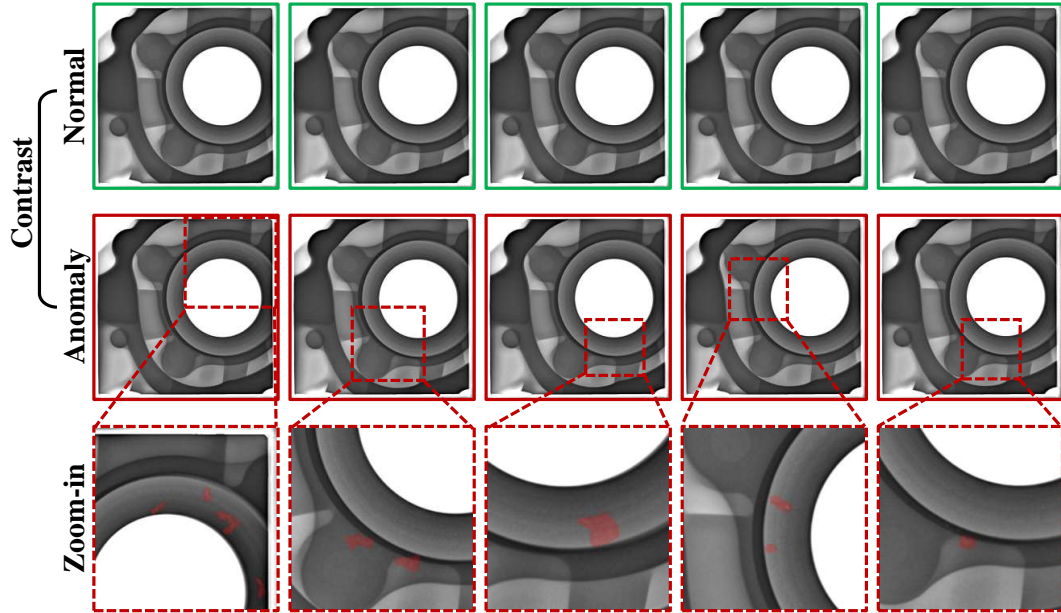


Figure 39: **Visualization of *Bearing 6* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

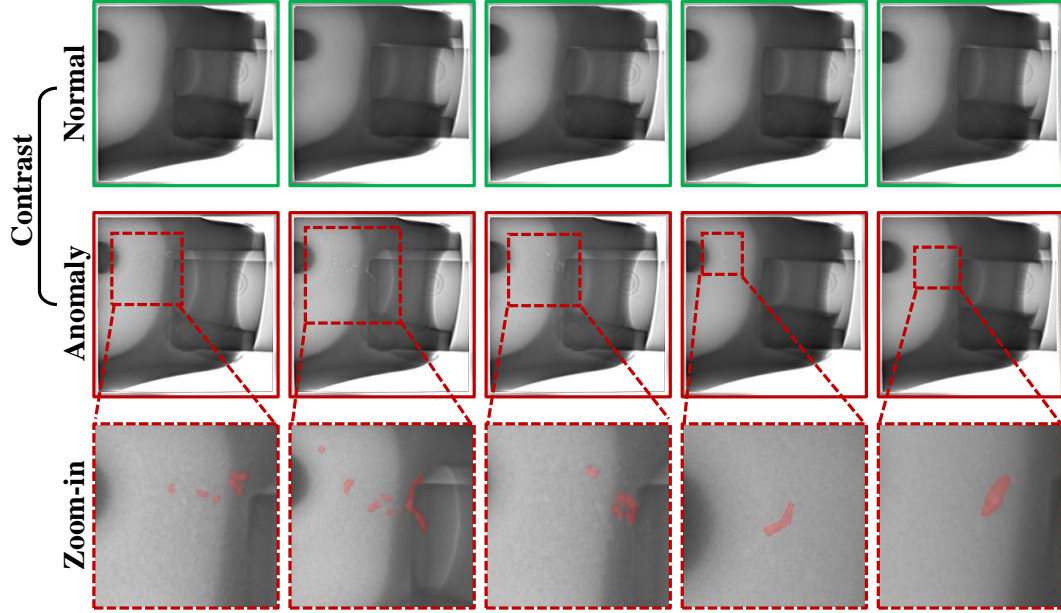


Figure 40: **Visualization of *Bearing 7* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

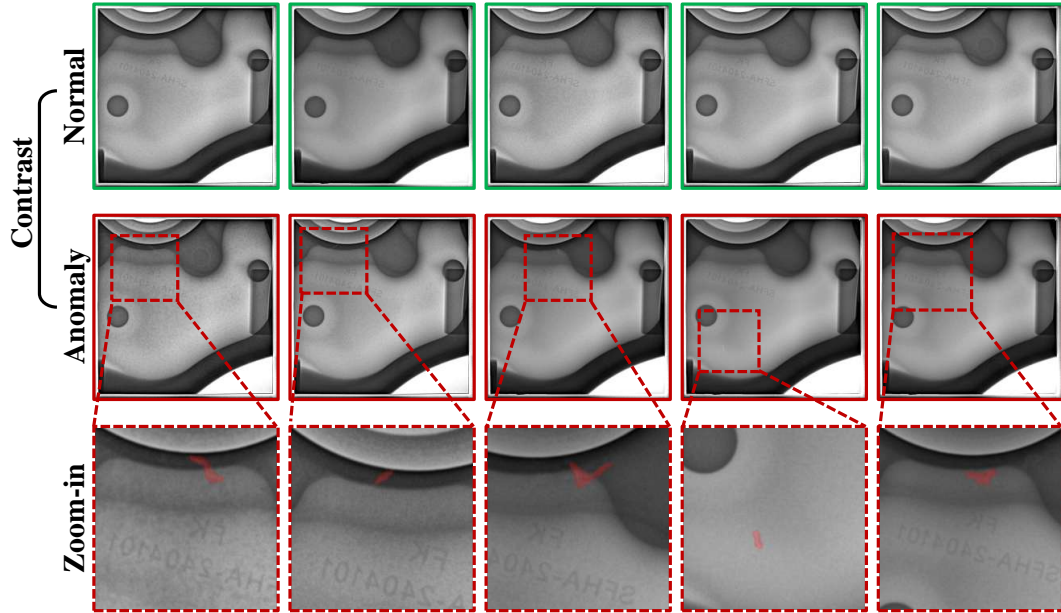


Figure 41: **Visualization of *Bearing 8* in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

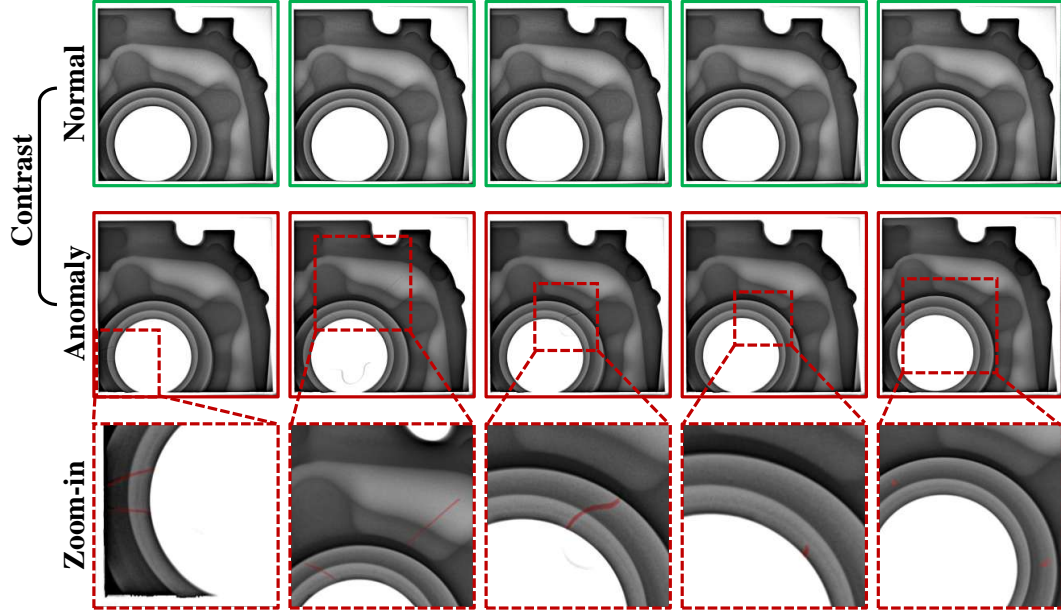


Figure 42: **Visualization of *Bearing 9* in ReinAD dataset.** Top row: Normal images with **green** borders; Middle row: Anomaly images with **red** borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in **red** masks.

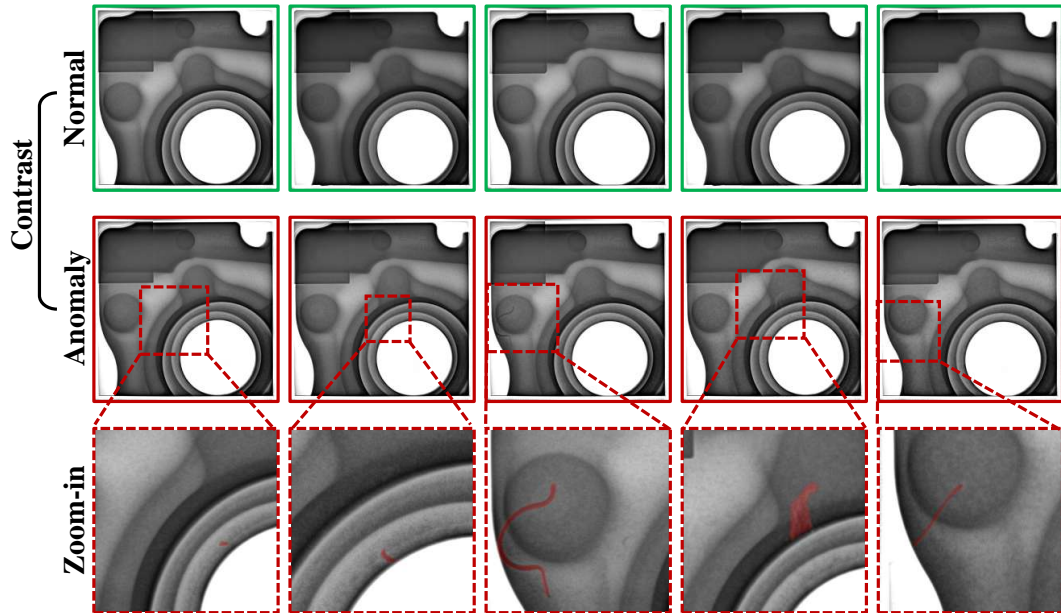


Figure 43: **Visualization of *Bearing 10* in ReinAD dataset.** Top row: Normal images with **green** borders; Middle row: Anomaly images with **red** borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in **red** masks.

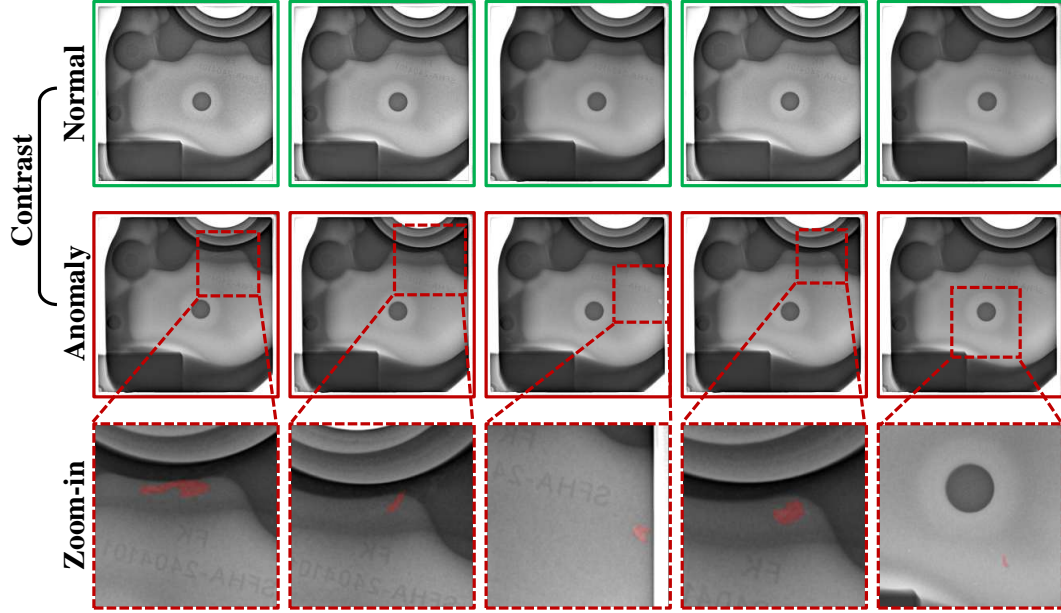


Figure 44: **Visualization of Bearing 11 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

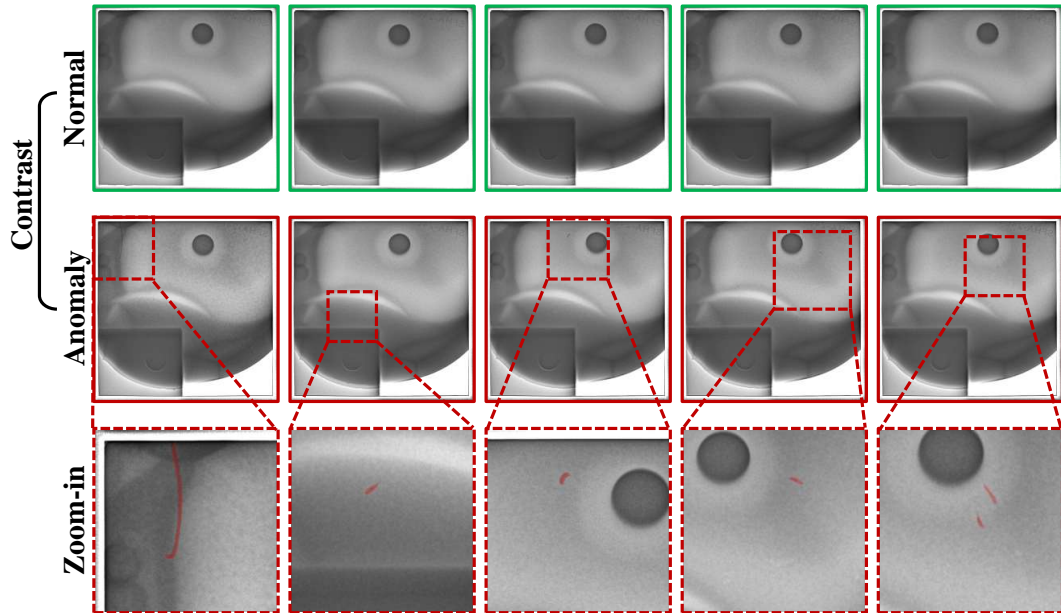


Figure 45: **Visualization of Bearing 12 in ReinAD dataset.** Top row: Normal images with green borders; Middle row: Anomaly images with red borders; Bottom row: Zoom-in patches of anomaly images, where anomalies are annotated in red masks.

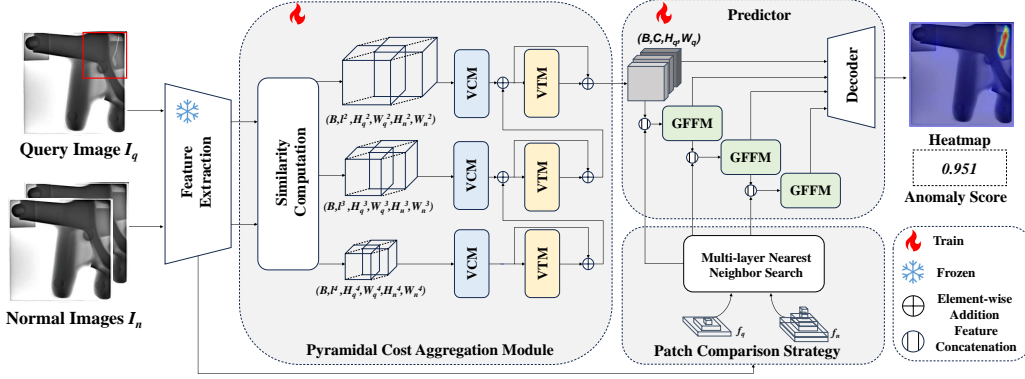


Figure 46: **Framework of our ReinADNet.** Given a query image and a set of reference images as input, a pretrained network extracts multi-scale features. The Pyramidal Cost Aggregation Module captures global point-to-point similarity between I_q and I_n , while the Patch Comparison Module captures local discrepancies. The predictor subsequently aggregates these discrepancy and similarity features to generate precise pixel-wise anomaly predictions.

C Additional Information of ReinADNet Method

C.1 Overview of Our Approach

Our approach localizes anomalies by identifying fine-grained differences between a query image and some reference normal images. As illustrated in Fig. 46, we simulate a contrastive process for anomaly detection on a given object class. The model simultaneously processes a query image I_q and a normal sample prompt I_n . We then select a pre-trained network as the backbone to extract multi-level features f_q^l from the query image and f_n^l from the normal image. We propose a coarse matching strategy to identify the most relevant regions within the normal feature maps that correspond to the query features, thereby obtaining coarse-matched feature representations. Building upon this, we compute global matching score matrices between query and normal feature maps across multiple hierarchical levels. These matching scores are subsequently refined through a cost aggregation module to enhance the robustness of the matching process. Furthermore, we design a top-down feature fusion module that integrates multi-scale matching scores with the coarse matching results to generate a refined anomaly score map. Finally, we extract the maximum anomaly values from the anomaly score map to determine the overall anomaly score for the image. During inference, one normal image from each class in the target dataset D_{tgt} is selected as a reference image to generate an anomaly localization heatmap and an anomaly score for each test image. We also provide a pseudocode to clarify how our method addresses misalignment, as illustrated in algorithm 1.

C.2 Pyramidal Cost Aggregation Module

To further capture semantic-level feature discrepancies, our model must not only search for local difference cues but also consider global context across images. It must establish correspondences between each query feature and its counterpart in a normal image, effectively learning semantic-level matches and thus detecting anomalies despite pixel shifts or viewpoint distortions. While semantic-matching literature commonly employs optical flow to achieve point-to-point image alignment by searching for corresponding semantic points, anomaly detection lacks flow supervision and cannot explicitly output such matches. Inspired by the central role of similarity in optical flow estimation, we adopt a cost aggregation scheme to perform global matching and implicitly guide the network to learn global similarity features. Specifically, after feature extraction, we obtain feature pairs $\{f_n^l, f_q^l\}_{l=1}^L$ across L distinct hierarchical levels. For each feature pair $\{f_n^l, f_q^l\}$ at a given level, we compute their initial cost maps:

$$C^l(i, j) = \frac{f_q^l(i) \cdot f_n^l(j)}{\|f_q^l(i)\| \|f_n^l(j)\|}, \quad (1)$$

where i, j represent the 2D spatial locations in the feature maps. By stacking the L^p cost maps computed at the same spatial scale, we obtain the initial cost maps across multiple scales $C^p \in$

Algorithm 1 Integrated Global-Local Strategy for misalignment.

Input: Query image feature map $\mathbf{f}_q \in \mathbb{R}^{H_q \times W_q \times C}$, reference normal feature map $\mathbf{f}_n \in \mathbb{R}^{H_n \times W_n \times C}$

Output: Anomaly heatmap $\mathbf{A} \in \mathbb{R}^{H_q \times W_q}$

Stage 1: The cost aggregation module for global point-to-point similarity

$\mathbf{C} = \text{zeros}(H_q, W_q, H_n, W_n)$

for each position (i, j) in \mathbf{f}_q **do**

for each position (k, l) in \mathbf{f}_n **do**

$$\mathbf{C}(i, j, k, l) = \frac{\mathbf{f}_q(i, j) \cdot \mathbf{f}_n(k, l)}{\|\mathbf{f}_q(i, j)\| \|\mathbf{f}_n(k, l)\|}$$

end for

end for

$\mathbf{M} = \text{Conv4d}(\mathbf{C})$

$\mathbf{A}_{agg} = \text{Swin4d}(\mathbf{M})$

Stage 2: The patch comparison module for local discrepancies

$\mathbf{f}_{dist} = \text{zeros}(H_q, W_q, C)$

for each position (i, j) in \mathbf{f}_q **do**

$\mathbf{f}_{close}(i, j) = \mathbf{f}_n(\text{argmin}_{k,l}(1 - \mathbf{C}(i, j, k, l)))$

$\mathbf{f}_{dist}(i, j) = \mathbf{f}_{close}(i, j) - \mathbf{f}_q(i, j)$

end for

Stage 3: Fusion and final prediction

$\mathbf{f}_{fusion} = \text{Swin2d}(\mathbf{A}_{agg} \oplus \mathbf{f}_{dist})$

$\mathbf{A} = \text{Conv2d}(\mathbf{f}_{fusion})$

return \mathbf{A}

$\mathbb{R}^{h_q \times w_q \times h_n \times w_n \times L^p}$, where (h_q, w_q) and (h_n, w_n) are the height and width of the query and normal feature maps, L^p represents the number of feature maps belonging to the same scale. The cost map C^p aggregates point-to-point similarity between each spatial location of the query features and the normal features, serving as the basis for subsequent matching.

Subsequently, we use a 4D convolutional network (VCM) to extract multi-level similarity features and utilize a 4D sliding window transformer (VTM) that transitions from coarse to fine granularity to aggregate these multi-level features, thereby obtaining a similarity feature map at the query feature scale. At each level l , the initial cost map C^p for that level is extracted using the VCM, resulting in the cost feature M^p for that level. Next, the VTM module is applied to further refine the features, yielding the feature map A^p , and a pyramid structure is employed to aggregate cost features from different levels, resulting in the final output:

$$M^p = \text{VCM}(C^p), \quad (2)$$

$$A^p = \text{VTM}(M^p), \quad (3)$$

$$A^{p-1} = \text{VTM}(M^p + \text{up}(A^p)), \quad (4)$$

where $M^p \in \mathbb{R}^{h_q \times w_q \times h_n \times w_n \times C}$, $A^p \in \mathbb{R}^{h_q \times w_q \times h_n \times w_n \times C}$, $\text{up}(\cdot)$ denotes bilinear upsampling. The aggregated final feature is $A^1 \in \mathbb{R}^{h_q \times w_q \times h_n \times w_n \times C}$. By taking the mean over h_n and w_n , we get the final feature $A \in \mathbb{R}^{h_q \times w_q \times C}$.

C.3 Patch-level Comparison Strategy

Several methods primarily leverage prototype learning paradigms to perform anomaly detection by deriving representative feature prototypes from training data and measuring their distances to test features. For instance, PaDiM utilizes the multivariate Gaussian distribution of normal features as the prototype, while PatchCore directly stores extracted normal features locally. Additionally, certain CLIP-based approaches involve prototype-based distance measurements between image features. This paradigm remains straightforward and effective in few-shot scenarios, as directly measuring

feature distances typically yields favorable results when inter-sample variance is limited. Thus, we introduce a Patch-level Comparison Strategy into our model. Inspired by prototype-based distance metrics, this module searches for the most dissimilar prototype feature corresponding to each patch at multiple feature scales and uses their differences as local discrepancy features.

Specifically, for each position (i, j) in the query feature map f_q^i , we compute the cosine distances between the feature vector at this position and all feature vectors in the normal feature map f_n . Then, we identify the position in f_n that has the smallest distance to the current patch in f_q , and obtain the set of closest features and difference features with the minimum distance to each corresponding part in f_q , denoted as f_{close} and f_{dist} . The detailed computation process is as follows:

$$f_{close} = f_n \left(\arg \min \left(1 - \frac{f_q \cdot f_n}{\|f_q\| \|f_n\|} \right) \right), \quad (5)$$

$$f_{dist}(i, j) = f_{close}(i, j) - f_q(i, j). \quad (6)$$

Intuitively, if the current patch in f_q is normal, it should closely match a corresponding patch in the normal feature set f_n , resulting in a very small minimum distance. While if the current patch contains an anomaly, it is unlikely to find a closely matching patch in f_n , leading to a relatively large minimum distance. For each patch in f_q , we find the patch in f_n with the smallest distance, and the concatenation between the two is used to obtain a preliminary difference feature map f_{dist} as a coarse measure of anomaly. To alleviate the limitations caused by fixed-size receptive fields, we integrate this module across multiple network layers, enabling it to capture local discrepancy features at various scales.

C.4 Predictor

We design a predictor to obtain the final anomaly heatmap. Given the aggregated similarity feature map A , we further aggregate it with the coarse residual features from different levels within the decoder using a pyramid structure. This helps to fully integrate the global similarity features with the local disparity features from each level, thereby enhancing detection performance. The specific process is as follows:

$$A^{l-1} = GFFM(A^l \oplus f_{dist}^l), \quad (7)$$

$$m^{l-1} = Conv(A^{l-1}), \quad (8)$$

where $GFFM(\cdot)$ denotes the 2D Swim Transformer, $Conv(\cdot)$ denotes a two-layer convolutional network decoder, \oplus denotes feature concatenation. A^{l-1} is updated from the previous global similarity feature A^l and the local residual feature f_{dist}^l . While decoding to obtain the heatmap m^{l-1} for level $l-1$, it is also further fused with the residual features f_{dist}^{l-1} from the last level to generate the new A^{l-2} . Finally, n heatmaps are output at different levels, and the average value of these n heatmaps is selected as the final prediction. The maximum anomaly value from the prediction map is chosen as the overall anomaly score of the image.

C.5 Training and Inference

Training. During the training phase, we freeze the parameters in the image encoder and only update the parameters of the other modules. In the data preparation phase, we first randomly select one normal sample from each object class to serve as the prompt image. Other normal and anomaly images are chosen as query images to simulate real-world normal/abnormal detection scenarios. For each normal query image, we generate a mask filled with 0 to indicate no defects. For anomaly query images, we use their respective masks, forming a complete supervised set G . Considering the imbalance in area between anomalous and normal regions, we select focal loss as the loss function, defined as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{x \in D_{src}} \mathcal{L}(S(x), G(x)), \quad (9)$$

where x represents each individual sample in D_{src} . The loss is computed between its anomaly heatmap $S(x)$ and the corresponding ground truth mask $G(x)$, and the mean of these losses across all samples is taken as the total loss.

Table 2: **Per-category anomaly detection results.** All models are trained and then tested on our ReinAD dataset under 1-shot setting. Metrics are image-level AUROC / AP / F_1 max. The best results are **bold**.

Categories	Patchcore	WinCLIP	InCTRL	ResAD	ReinADNet (Ours)
Bearing 1	46.6/8.0/26.7	75.0/6.8/15.1	54.5/2.5/6.6	80.4/31.1/45.5	99.3/72.7/80.0
Cable 1	75.1/57.2/60.8	72.0/55.9/62.8	50.5/36.9/50.8	83.7/61.6/73.0	80.3/65.8/69.9
Cable 3	87.9/80.3/80.3	87.6/78.6/78.0	58.4/42.6/57.9	94.0/86.3/87.4	90.9/87.6/83.2
Led 1	59.3/31.0/32.6	82.1/63.7/60.1	72.2/45.2/42.5	55.0/20.4/30.4	76.6/44.8/48.1
Led 3	73.8/29.2/42.3	71.2/39.2/40.8	60.6/27.3/30.9	79.4/44.7/54.6	77.1/50.7/48.8
Lens 2	52.8/53.8/66.1	60.6/62.9/66.6	46.7/45.6/66.2	58.2/54.0/67.8	70.5/72.2/69.6
Metal piece 1	52.8/35.4/55.4	64.9/45.8/55.0	71.1/51.3/59.1	34.1/26.4/49.8	41.3/32.2/49.5
Metal piece 2	63.3/34.5/48.5	67.0/36.6/47.0	48.7/24.0/41.2	71.9/44.6/51.1	72.5/46.1/52.2
Metal piece 4	47.5/33.5/52.7	72.6/63.9/59.0	64.4/47.5/54.9	64.5/54.2/54.6	90.3/89.4/80.9
Metal piece 5	53.8/37.8/53.2	55.6/45.4/53.0	43.8/32.0/53.0	58.5/43.3/55.4	72.2/64.2/59.8
Metal piece 7	48.4/32.0/54.7	61.2/44.0/52.7	78.6/61.2/64.8	77.2/59.3/64.9	80.6/71.0/65.7
Metal piece 8	44.6/96.7/ 98.5	53.7/97.4/ 98.5	42.3/96.6/ 98.5	55.7/97.7/ 98.5	57.5/98.0/98.5
PCB solder 1	94.9/97.7/92.9	87.2/94.5/88.0	0.0/49.3/81.3	88.5/95.3/88.0	67.9/85.0/86.7
PCB solder 2	66.9/88.1/88.0	80.5/94.4/88.0	74.7/92.7/86.3	81.8/92.3/ 95.6	93.5/98.0/93.6
Piston ring 1	48.3/20.8/37.0	59.9/29.2/39.2	50.8/24.2/36.5	62.1/31.2/39.8	55.1/24.5/37.8
Piston ring 3	56.9/40.0/45.7	64.9/41.8/49.8	54.5/35.0/45.6	56.2/35.4/46.3	57.9/34.8/46.7
Plastic box	60.7/ 97.0/97.8	54.9/95.5/ 97.8	12.7/90.1/97.6	61.4/96.5/ 97.8	64.1/96.7/97.8
Plastic cover 1	62.9/92.5/94.8	71.4/94.8/94.8	41.8/88.1/93.7	72.0/94.9/ 95.3	76.8/95.9/93.7
Profile surface 1	62.0/56.6/67.8	66.9/65.9/68.3	66.3/63.8/69.0	65.0/60.6/ 69.1	58.3/55.8/67.4
Thread	40.0/29.5/50.0	62.3/42.1/52.7	55.0/38.5/50.0	43.8/32.6/50.0	48.5/32.8/50.0
Wafer 1	78.7/80.6/71.0	83.4/81.1/76.4	65.6/63.4/65.4	78.5/ 81.4/69.3	71.0/70.0/68.9
Wafer 2	50.1/20.0/34.2	56.9/29.4/34.3	58.9/29.6/35.0	51.6/22.2/34.2	63.7/30.4/38.0
Average	60.3/52.4/61.4	68.7/59.5/62.6	53.3/49.4/58.5	67.0/57.5/64.5	71.2/64.5/67.6

Inference. At inference, for each class, we sample a few normal samples as reference. Given a query image for testing, we input both the query image and the reference samples into the network to obtain the anomaly heatmap S for the query image, representing patch-level anomaly scores. Based on previous methods, we apply filtering to S and take the maximum anomaly value over the entire image as the image-level anomaly score.

D Additional Information of Experiments

D.1 Detailed Quantitative Results for Each Category

Tab. 2 and Tab. 3 include the per-category 1-shot quantitative results. Our ReinADNet exhibits significant advantages in handling the following challenging scenarios: anomalies requiring normal-anomaly contrast to detect (*e.g.*, metal piece and LED), misaligned samples (*e.g.*, wafer and PCB solder), fine-grained anomalies (*e.g.*, plastic box and plastic cover), and multi-class anomalies (*e.g.*, bearing and lens).

D.2 Additional Qualitative Results

Fig. 47 to Fig. 49 show additional qualitative results of all test categories under 1-shot setting. Most state-of-the-art methods fail to generate good anomaly localization maps for unseen categories, due to many false positives in normal regions. However, our method effectively reduces false positives in normal regions and locate anomalies more accurately. The visual results confirm our ReinADNet’s superiority in handling both misalignment and fine-grained anomalies.

Table 3: **Per-category anomaly localization results.** All models are trained and then tested on our ReinAD dataset under 1-shot setting. Metrics are pixel-level AUROC / AP / F_1 max. The best results are **bold**.

Categories	Patchcore	WinCLIP	InCTRL	ResAD	ReinADNet (Ours)
Bearing 1	88.5/1.6/5.8	92.7/1.1/5.3	-/-	98.3/10.3/25.3	99.0/36.3/41.8
Cable 1	97.9/ 16.9 /22.1	95.2/4.0/9.9	-/-	98.5 /8.8/18.1	97.3/13.9/ 23.8
Cable 3	98.6/21.6/30.5	99.1/17.1/27.1	-/-	99.7 /41.4/47.5	99.3/ 49.0/51.0
Led 1	94.0/9.2 /17.0	85.4/6.2/12.9	-/-	83.9/0.3/0.7	88.6/7.8/ 17.2
Led 3	92.9/0.6/1.7	91.7/4.7/12.3	-/-	98.1/13.5/25.8	92.5/3.0/8.4
Lens 2	96.3/22.3/29.0	97.6/37.8/43.0	-/-	98.8/16.6/28.3	99.2/60.0/58.5
Metal piece 1	70.8/0.0/0.0	83.5 /0.3/0.8	-/-	76.0/0.2/0.5	83.4/ 0.4/1.6
Metal piece 2	89.7/1.5/3.6	95.2/ 9.5/17.0	-/-	96.2 /8.0/13.8	91.6/4.4/8.7
Metal piece 4	65.1/0.4/0.9	92.8/16.8/25.3	-/-	90.1/11.2/16.5	98.4/59.5/60.3
Metal piece 5	53.3/2.4/4.9	52.5/3.4/6.8	-/-	55.1/4.2/10.0	57.1/7.0/12.1
Metal piece 7	79.6/0.8/2.6	86.7/7.5/16.1	-/-	93.7/3.2/6.4	97.1/20.3/27.8
Metal piece 8	61.4/2.9/7.0	58.4/2.7/5.8	-/-	78.3 /6.0/ 11.2	73.2/ 6.2 /10.0
PCB solder 1	91.0/1.8/4.2	92.9/4.7/12.2	-/-	97.6/12.9/17.1	96.4/4.8/9.1
PCB solder 2	93.6/ 26.7 /31.0	93.8/9.0/18.0	-/-	98.1 /25.8/ 32.5	95.2/23.1/32.2
Piston ring 1	87.7/0.0/0.0	93.5/0.1/0.2	-/-	95.9/ 0.5/2.5	98.4 /0.4/1.1
Piston ring 3	68.3/0.0/0.0	92.9/0.1/0.7	-/-	94.1/0.1/0.7	97.7/1.3/4.0
Plastic box	76.1/0.0/0.0	90.5/0.0/0.1	-/-	85.1/0.0/0.3	94.0/0.6/3.7
Plastic cover 1	91.2/0.0/0.2	92.6/3.7/15.6	-/-	93.9/0.8/5.1	98.1/7.8/17.0
Profile surface 1	61.1/0.8/2.1	54.1/0.9/3.3	-/-	62.2/1.0/3.2	73.2/2.7/7.4
Thread	54.8/0.2/0.5	76.1/5.0/12.1	-/-	85.7 /2.3/7.9	82.5/ 5.1/14.6
Wafer 1	96.2/58.9/63.2	82.2/30.4/35.6	-/-	96.3/61.1/70.1	72.9/24.2/30.2
Wafer 2	94.1/0.1/0.3	91.5/4.2/ 11.3	-/-	95.2/1.1/3.6	98.9/4.6/8.0
Average	81.9/7.7/10.3	85.9/7.7/13.2	-/-	89.6/10.4/15.8	90.2/15.6/20.4

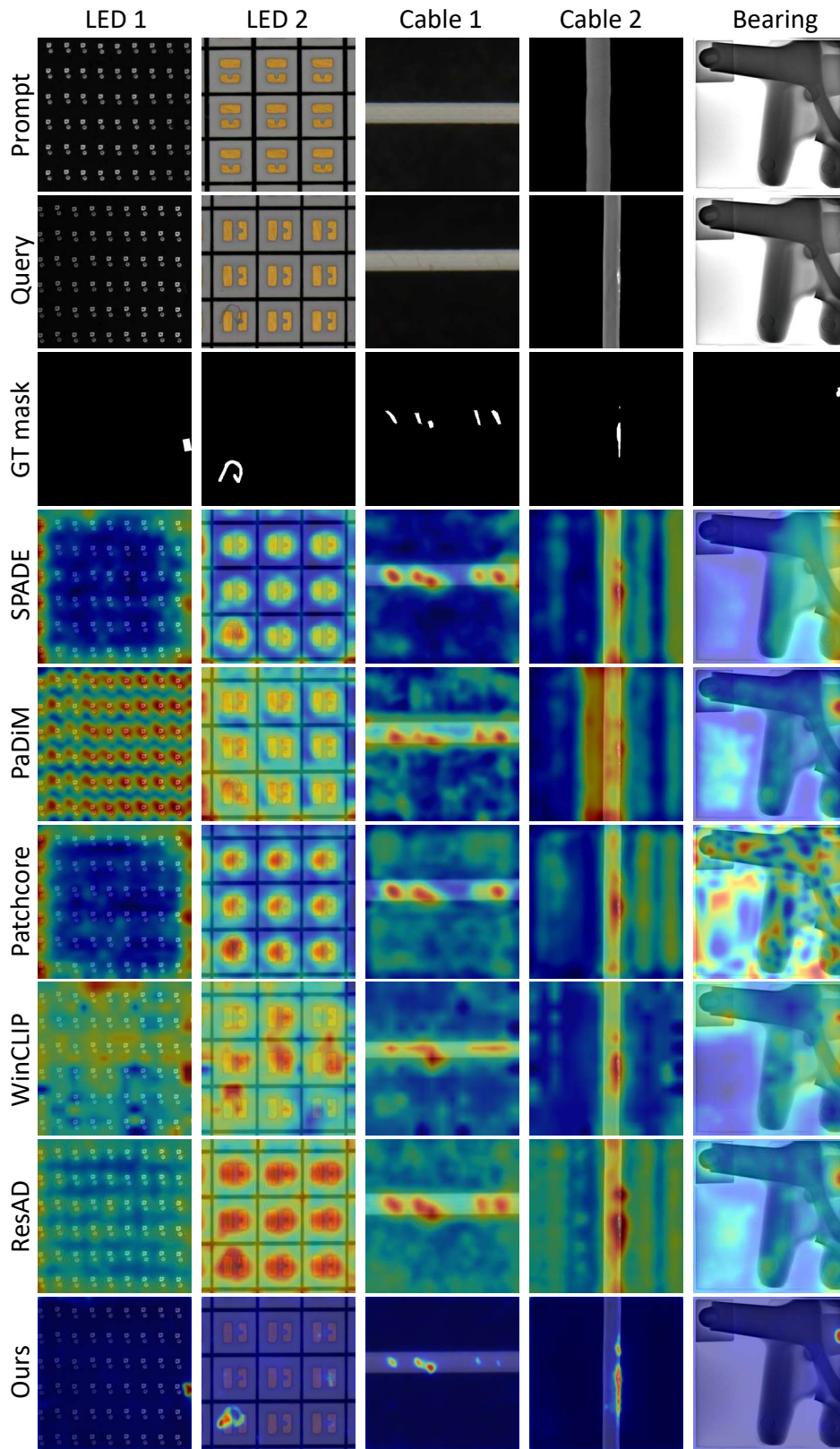


Figure 47: Qualitative results.

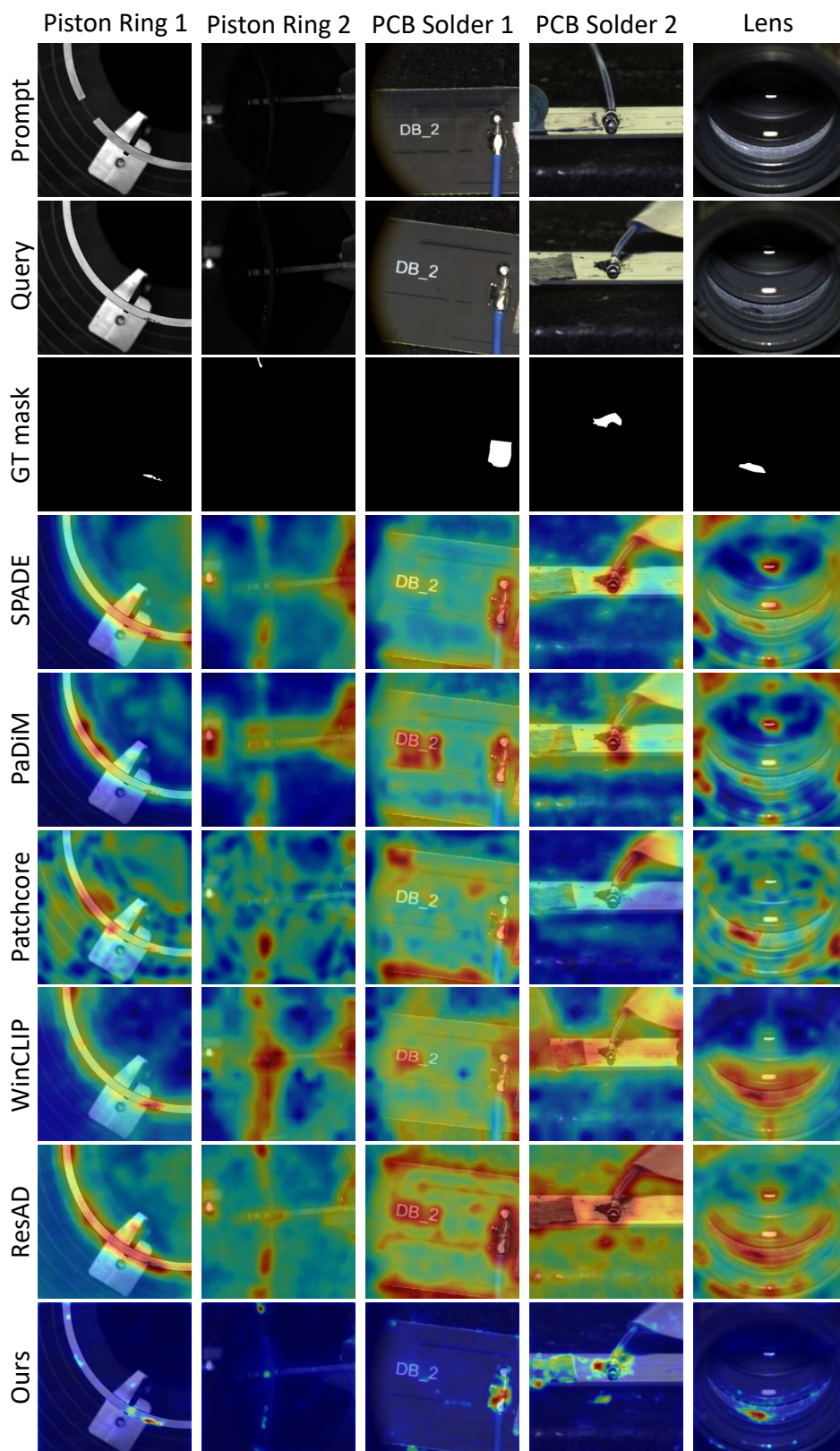


Figure 48: **Qualitative results.**

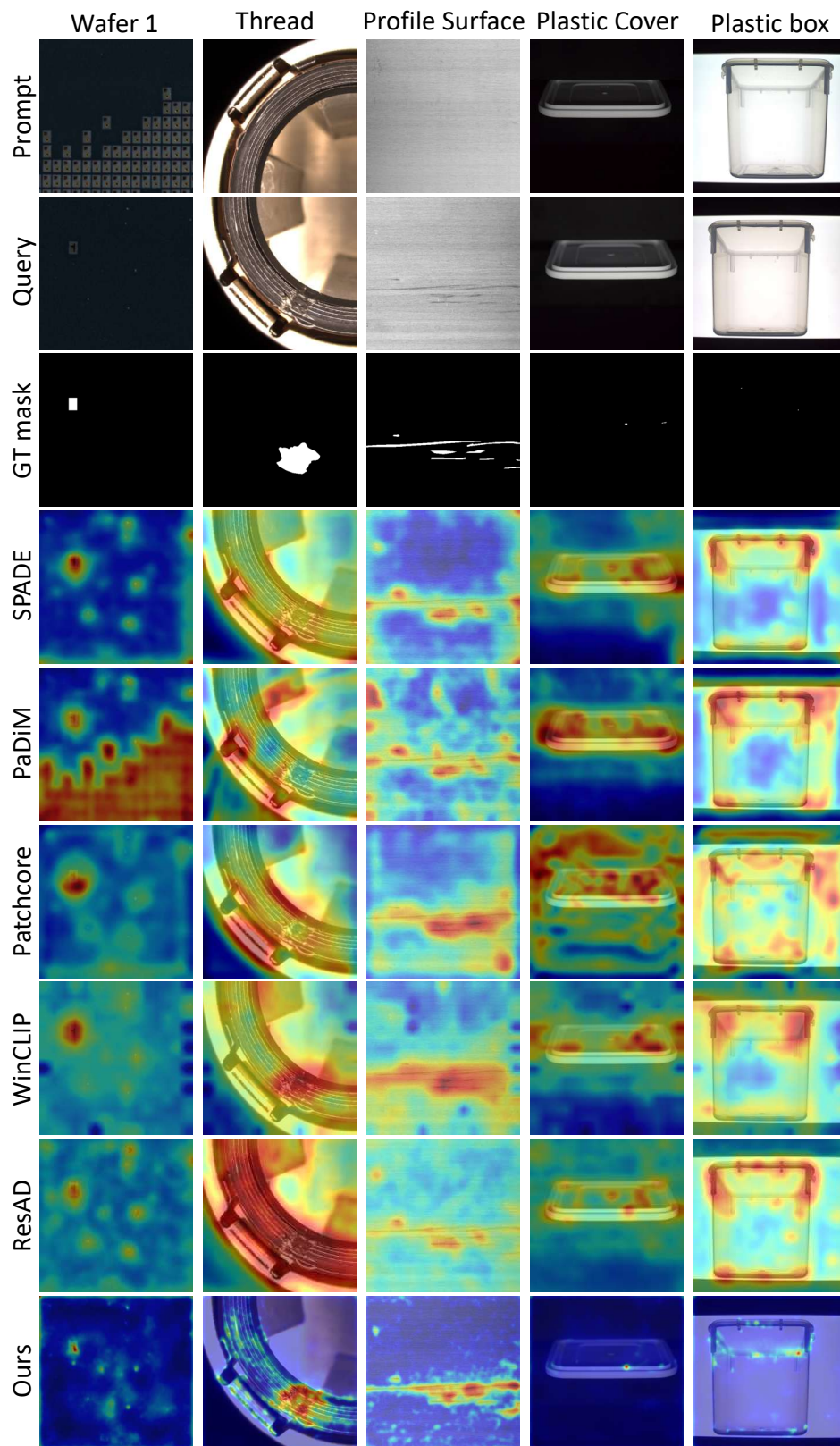


Figure 49: **Qualitative results.**